UNIVERSITÀ DEGLI STUDI DI SIENA

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

---

Ph.D. Information Engineering and Science

Contributions to ceers, logical depth, algorithmic randomness and their applications

Supervisor:
Prof. Andrea Sorbi

Candidate:
Valentino Delle Rose

XXXIV Cycle

“Tra quello che sappiamo e quello che ci inventiamo, sempre ce la caviamo.”
(Motto della famiglia Delle Rose)

Ai miei genitori Graziella e Maurizio, che mi hanno sempre accompagnato verso i miei sogni.

Acknowledgements

Time has flown by and I have already reached the time to thank all the people who have accompanied me on this journey.

First, I want to thank my supervisor, prof. Andrea Sorbi, simply for being the special person he has been in these three years. He has been the most helpful and patient supervisor I could have asked for and I will always be thankful to have had the opportunity to work with him.

I also want to express my gratitude to all the wonderful colleagues I have worked together during my PhD studies.

There are several reasons for which I want to thank prof. Laurent Bienvenu, surely way more than the ones I could write here. Despite the distance, made even more difficult to overcome by the pandemic situation we have all had to face in the last years, he has been a present and forbearing mentor and working with him has been a truly enriching experience in every respect. I am especially grateful to him for hosting me in Bordeaux, in March 2021: the month I have spent there has been very important for me.

Next, I want to thank Dr Wolfgang Merkle. This is already the second thesis in which I thank him, and he really deserves it. For a few years now, he has played a big role in my mathematical education and research work and he is one of the main reasons I can consider Heidelberg as my second university. I am particularly thankful to him for hosting me several times there again during my PhD studies.

I also want to express my sincere gratitude to Dr Luca San Mauro, for all the precious time we have spent working together: I have really learned a lot from him and I am thankful for all the insightful discussions we had in the last three years.

Finally, I want to offer my special thanks to Dr Tomasz Steifer, who has been both a wonderful colleague and a helpful friend to me. I am truly grateful to have met him and worked with him.

I really wish that the professional and personal bond I have established with each of them will continue for a long time to come.

Next, I want to thank all my friends for their love and support, without which these years would have been much harder. So, I want to thank my colleagues and friends from San Niccolò, Chiara, Ilaria and Luca: thank you for all the time spent together studying and having fun, especially for our summer holidays at the beach, which I will miss a lot. Special thanks to Paola, Eva, Gasta, Rocco, Lele, Frenz and all the rest of the ESN Siena GES family: I have been so lucky to have found such trustworthy friends here in Siena. And to Daniela, for being almost my deputy mum and always keeping me safe. Also, to Carlotta, Maria and Damiano, for having the strength to be my lifelong friends: thank you for always being close to me, even if we live far apart. And to Chiara, for loving me as I am and always supporting me, thank you from the bottom of my heart.

Last, but not least, I want to express all my gratitude to my loving family: without their unconditional help and support nothing I have done would have been possible. I hope they are at least as proud of me as I am of them.

Valentino

Abstract

This thesis collects some contributions to different fields of computability theory and algorithmic randomness, on which I have been focusing during my PhD studies.

The first line of research, on which I have worked jointly with Luca San Mauro and Andrea Sorbi, concerns computable reducibility on the equivalence relations on the set of natural numbers. Extending Ershov's category-theoretic approach (see [40]), we have investigated various properties of the category of equivalence relations on $\mathbb{N}$, where the morphisms from an equivalence relation R to another equivalence relation S are those maps from R-equivalence classes to S-equivalence classes, which are induced by computable functions. We have also studied some important full subcategories of $\mathbb{E}\mathrm{q}$, such that the ones of c.e. equivalence relations, which we often referred to simply as *ceers*, and of co-c.e. relations.

Moreover, we have studied the "expressiveness" of certain classes of effectively presented algebraic structures using the tools of computable reducibility: namely, we have studied which ceers lie in the same degree (with respect to the degree structure induced by computable reducibility) of the word problem of some member of various classes of familiar algebraic structures, such as semigroups, groups and rings. Our main result, which answers an open question from [45], is that, while in every degree there is the word problem of some c.e. semigroup, there are ceers which are not bi-reducible to the word problem of any finitely generated semigroup: in fact, it is even possible to identify a natural computability-theoretic property (which we have called *hyperdarkness*) preventing ceers from being in the same degree of the word problem of any finitely generated semigroup.

The second project, joint with Laurent Bienvenu and Wolfgang Merkle, focuses on logical depth. This notion has been introduced by Bennett in [9] and aims to capture the intuitive idea of the internal organization of

information. In particular, we have studied how depth relativizes to various classes of oracles, in order to better understand which oracles do actually help in organizing information: the main results are that the class of deep sequences with respect to the halting set is incomparable (with respect to the inclusion) with the corresponding unrelativized class, while the class of deep sets relative to any Martin-Löf random sequence strictly contains the unrelativized one. A consequence of our results is that we slightly strengthen a result in [7], stating that every PA-complete degree is the join of two ML-random degrees: in fact, we show that every DNC_2 function is truth-table-equivalent to the join of two Martin-Löf random sets.

Finally, in the last project, which is joint work with Laurent Bienvenu and Tomasz Steifer, we have compared deterministic forecasting schemes against probabilistic ones, using the toolkit of algorithmic randomness and, in particular, the notion of martingales. We have introduced a new notion in the "randomness zoo": we call a sequence X *almost everywhere computably random* if, for almost every sequence Y (i.e. up to a null class) X is computably random relatively to Y. Notice that this approach is indeed equivalent to consider probabilistically computable martingales, by simply assuming that Y has been drawn at random in advance. Then, using the so-called *fireworks technique* (see, e.g., [16, Section 1.4]), we have built a partial computable sequence (roughly speaking, a sequence on which no deterministic martingale succeeds) which is not almost everywhere computably random, hence proving that probabilistic martingales are actually stronger than deterministic ones. It is worth noticing that this is a quite unusual result in computability theory, starkly contrasting the classical result that the sequences which can be computed by some probabilistic algorithm with positive probability coincide with the deterministically computable ones [31].

Contents

Acknowledgements . . . iii
Abstract . . . v

Outline of the thesis **x**

I Contributions to the theory of computable reducibility on equivalence relations on $\mathbb{N}$ **1**

Introduction **2**

1 Preliminaries **4**
1.1 Computable reducibility among equivalence relations on $\mathbb{N}$ . . . 4
1.2 Finite, light and dark ceers . . . 5
1.3 Isomorphisms and strong isomorphisms of ceers . . . 9
1.4 Some universal ceers . . . 11

2 On the category of equivalence relations **14**
2.1 Preliminaries on category theory . . . 15
2.2 The category of equivalence relations on $\mathbb{N}$ . . . 20
2.2.1 Monomorphisms and epimorphisms of equivalence relations, ceers and coceers . . . 22
2.2.2 Products and coproducts in $\mathbb{E}\mathrm{q}$ . . . 26
2.2.3 Equalizers and coequalizers . . . 28
2.2.4 Subcategories of $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ and closure under binary coproducts and coequalizers . . . 31
2.3 Conclusion . . . 32

3 The expressive power of algebraic structures by means of computable reducibility **33**
3.1 C.e. algebras 34
3.2 Classes of algebras that are complete for the ceers 36
3.2.1 The word problem as a ceer on terms, or as a ceer on the free algebra 38
3.2.2 Semigroups 39
3.2.3 Monoids 40
3.3 Classes of algebras that are not complete for the ceers 40
3.3.1 Semigroups 40
3.3.2 Monoids 41
3.4 On finitely presented semigroups and a question of Gao and Gerdes 42
3.5 Transversals of word problems of finitely generated semigroups 50
3.5.1 More immunity and darkness notions 50
3.5.2 Π^0_1 classes and $\equiv$-realizability by word problems of finitely generated semigroups 51
3.5.3 The hyperdark ceers 56
3.6 Classes of algebras $\simeq_s$-realizing provable equivalence of Peano Arithmetic 62
3.7 Conclusion 68

II Contributions to the theory of logical depth and algorithmic randomness **70**

Introduction **71**

4 Relativization of Bennett's notion of depth **74**
4.1 Preliminaries 76
4.1.1 Kolmogorov complexity 77
4.1.2 Lower-semicomputable discrete semimeasures 79
4.1.3 Martin-Löf randomness 80
4.1.4 Depth 82
4.2 Relativized depth 91
4.2.1 $\emptyset'$-depth 95
4.2.2 Depth relative to ML-random oracles 97
4.2.3 A digression on a result about PA-complete degrees . . 103

4.2.4 An open question about K-trivial oracles 106
4.3 Turing-relativized depth 107

5 On the comparison between deterministic and probabilistic forecasting schemes 111
5.1 Preliminaries 113
5.1.1 Computable and partial computable randomness 113
5.1.2 Probabilistic martingales 115
5.2 Turing degrees of a.e.CR sequences 117
5.3 The main construction 122
5.3.1 Defeating finitely many martingales 123
5.3.2 Defeating all partial computable martingales 124
5.3.3 Fireworks 125
5.4 Conclusion and open questions 129

Bibliography 131

Outline of the thesis

The underlying theme of this thesis is how the computational properties of mathematical and logical objects relate to their expressive power, the complexity of the information these objects encode and their usefulness in helping solving certain problems. Naturally, this is a broad question, which can be investigated from different perspectives and studied according to various approaches. In this thesis, we consider two main approaches to this wide and general question: the first one is strongly related to positive structures and c.e. equivalence relations on the natural numbers, while the second one makes an essential use of tools from algorithmic information theory and algorithmic randomness. Therefore, the thesis is divided into two parts, which we briefly describe below.

The central notion of Part I is computable reducibility on equivalence relations on the set $\mathbb{N}$ of natural numbers.

- In Chapter 1, we review useful concepts and results from the literature on computable reducibility, on which this first part of the dissertation is based.

- Chapter 2 concerns some category-theoretic properties of computable reducibility: we investigate the category $\mathbb{E}\mathrm{q}$ of equivalence relations on $\mathbb{N}$ and their full subcategories $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ and $\mathbb{E}\mathrm{q}(\Pi^0_1)$ of, respectively, c.e. and co-c.e. equivalence relations. This chapter is based on the paper [34].

- In Chapter 3, we investigate the expressive power of various familiar classes of algebraic structures using computable reducibility as main tool: indeed, we look at the problem of which ceers can be realized by word problems of computably enumerable structures (such as c.e. semigroups, groups, and rings), where being realized means to fall in the same degree (with respect to the degree structure induced by computable reducibility, or by some stronger variant). Most of the results collected in this chapter have been published in [33].

In Part II, instead, we mainly use the toolkit provided by the theory of algorithmic randomness.

- In Chapter 4, we investigate the relativization of Bennett's notion of logical depth. In fact, we propose two possible definitions of depth relative to an oracle, although one receives here more attention, being, in our opinion, more interesting as much closer to Bennett's original definition from a "philosophical" point of view. This chapter is mainly based on [13].

- Finally, in Chapter 5, we develop the notion of *almost everywhere computable randomness*, in order to compare the strength of probabilistic forecasting schemes against deterministic ones. This chapter is based on [14].

Part I

Contributions to the theory of computable reducibility on equivalence relations on $\mathbb{N}$

Introduction

In recent years there has been a growing interest in the investigation of the relative complexity of equivalence relations on the set $\mathbb{N}$ of natural numbers by means of the so called *computable reducibility*: given equivalence relations R, S on $\mathbb{N}$, we say that R is *computably reducible* to S if there exists a computable function f such that for all pairs $x, y \in \mathbb{N}$, $x \; R \; y$ if and only if $f(x) \; S \; f(y)$.

The first systematic study of this reducibility is due to Ershov ([40]), as an alternative way to look at monomorphisms in the category of numberings.

In Chapter 1, we review some definitions and known facts about computable reducibility, which will be useful in the following chapters.

In Chapter 2, extending Ershov's category-theoretic approach, we investigate some properties of the category $\mathbb{E}\mathrm{q}$ of equivalence relations, where a morphism between two equivalence relations R, S is a mapping from the set of R-equivalence classes to that of S-equivalence classes, which is induced by a computable function. We also consider its full subcategories $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ of computably enumerable equivalence relations (also called *ceers*), $\mathbb{E}\mathrm{q}(\Pi^0_1)$ of co-computably enumerable equivalence relations (or *coceers*) and $\mathbb{E}\mathrm{q}(\mathrm{Dark}^*)$ whose objects are the so-called dark ceers plus the ceers with finitely many equivalence classes. Although in all these categories the monomorphisms coincide with the injective morphisms, we show that in $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ the epimorphisms coincide with the surjective morphisms, but in $\mathbb{E}\mathrm{q}(\Pi^0_1)$ there are epimorphisms that are not surjective. Moreover, $\mathbb{E}\mathrm{q}$, $\mathbb{E}\mathrm{q}(\Sigma^0_1)$, and $\mathbb{E}\mathrm{q}(\mathrm{Dark}^*)$ are closed under finite products, binary coproducts, and coequalizers. On the other hand, we show that $\mathbb{E}\mathrm{q}(\Pi^0_1)$ does not always have coequalizers. The results collected in this chapter have been published in [34].

Finally, in Chapter 3, we use computable reducibility (and stronger variants) to study the expressive power of effectively presented algebraic structures. More precisely, we address the issue as to which ceers can be realized by word problems of computably enumerable (or, simply, c.e.) structures (such as c.e. semigroups, groups, and rings), where being realized means to fall in the same degree (with respect to the degree structure induced by computable reducibility), or in the same isomorphism type (with the isomorphism induced by a computable function), or in the same strong isomorphism type (with the isomorphism induced by a computable permutation of the natural numbers). We observe for instance that every ceer is isomorphic to the word problem of some c.e. semigroup, but (answering a question in [45]) not every ceer is in the same reducibility degree of the word problem of some finitely presented semigroup, nor is it in the same reducibility degree of some non-periodic semigroup. Indeed, we identify a whole class of ceers which cannot be realized by the word problem of any finitely generated semigroup, namely those ceers whose infinite transversal (i.e. infinite sets such that all pairs of distinct elements are not equivalent) are all hyperimmune. We also show that the ceer provided by provable equivalence of Peano Arithmetic is in the same strong isomorphism type as the word problem of some non-commutative and non-Boolean c.e. ring. Most of the results presented in this chapter have appeared in [33].

Chapter 1

Preliminaries

In this chapter, we review some useful terminology and known facts on the computable reducibility among equivalence relations on the natural numbers. The reader is referred to [89] for any unexplained notion from computability theory.

1.1 Computable reducibility among equivalence relations on $\mathbb{N}$

A popular way to compare the relative complexity of two equivalence relations R, S on the set $\mathbb{N}$ of natural numbers is by mean of the so-called *computable reducibility*.

Definition 1.1.1. Let R, S be equivalence relations on $\mathbb{N}$. We say that R is *computably reducibile* (or simply *reducible*) to S (and write $R \leq S$) if there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that

$$(\forall x, y) \left[x \; R \; y \;\; \Leftrightarrow \;\; f(x) \; S \; f(y) \right].$$

We write $R < S$ to mean that $R \leq S$ but $R \not\leq S$.

If both $R \leq S$ and $S \leq R$, we say that R and S are *bi-reducible* to each other (denoted by $R \equiv S$). The class of equivalence relations that are bi-reducible to some equivalence relation R is called the *degree* of R.

The above reducibility is stronger than usual m-reducibility among subsets of $\mathbb{N}$. Indeed, first notice that, if $R \leq S$ via some reduction f, then

$R \leq_m S$ via the computable function $(x, y) \mapsto \langle f(x), f(y)\rangle$, where $\langle \cdot, \cdot \rangle$ denotes the Cantor pairing function (see, for instance, [89, p. xxxii]).

To give a counterexample to the converse implication, it is convenient to introduce a special class of equivalence relations, called *unidimensional* after [45].

Definition 1.1.2. Given a set $X \subseteq \mathbb{N}$ define the equivalence relation R_X by

$$u \; R_X \; v \;\Leftrightarrow\; u = v \text{ or } u, v \in X,$$

namely the equivalence relation having X has its only equivalence class which is not a singleton. R_X is called the *unidimensional ceer generated by* X.

Then, for any set A, $A \leq_1 R_A$ via the reduction $f(x) = (x, a)$, where a is some fixed element of A. It is also clear that $R_A \leq R_B$ implies $A \leq_m B$.

Let us denote by Id the identity relation. Id is clearly computable, hence $\mathrm{Id} <_1 X \leq_1 R_X$ for any uncomputable set X. On the other hand, there are c.e. sets X so that Id and R_X forms incomparable degrees under computable reducibility. To see this, we need to recall the classical notions of *immunity* and *simplicity* for sets.

Definition 1.1.3. A set is *immune* if it is infinite but contains no infinite c.e. subset. A c.e. set is *simple* if its complement is immune.

Clearly $R_X \leq \mathrm{Id} \equiv R_\emptyset$ implies that $X \leq_m \emptyset$. But then X would be empty, whereas $\emptyset$ is obviously not simple. On the other hand, let $\mathrm{Id} \leq R_X$ via some computable function f. First we notice that $X \cap \mathrm{range}(f)$ contains at most one element. Indeed, assume that there are x, y with $f(x) \neq f(y)$ and $f(x), f(y) \in X$: then $f(x) \; R_X \; f(y)$, implying that $x = y$, a contradiction. But then $\mathrm{range}(f) \smallsetminus X$ is an infinite c.e. subset of $\overline{X}$, meaning that X is not simple. Hence, Id and R_X are incomparable.

1.2 Finite, light and dark ceers

We say that an equivalence relation R is *finite* if it has finitely many equivalence relations, *infinite* otherwise. Finite ceers are easy to classify. In fact, it is straightforward to show that they are all computable. It is also obvious that all ceers having exactly n equivalence classes are bi-reducible to each other: we denote their degree by Id_n. The prototype of ceer in Id_n is the congruence modulo n (where $x \equiv y \mod n$ means that $x = qn + y$ for some integer q).

For the rest of the section, we only consider computably enumerable equivalence relations, which we also call simply *ceers*. The following notions have been proposed in [5].

Definition 1.2.1. Let R be an infinite ceer. R is called *light* if $\mathrm{Id} \leq R$, *dark* otherwise.

Thus, ceers can be partitioned into finite, light and dark ceers. Moreover, it is clear from the definition that those notions are degree invariant (meaning that a degree contains a light ceer if and only if it contains only light ceers, and similarly for dark ceers). We sometimes denote the class of (degrees of) finite ceers by $\mathcal{F}$, that of light ceers with **Light**, and that of dark ceers with **Dark**. Finally, we also use the notation $\mathbf{Dark}^* = \mathbf{Dark} \cup \mathcal{F}$.

Light and dark ceers can be characterized using the notion of *transversal*.

Definition 1.2.2. Let R be an equivalence relation: a *transversal* of R is a set T such that, for every pair of distinct elements $x, y \in T$, $x \not R y$.

Light ceers can be characterized as those ceers having an infinite c.e. transversal. Indeed, assume that R is a light ceer, so that $\mathrm{Id} \leq R$ via some computable function f: then $range(f)$ is an infinite c.e. transversal of R. Conversely, if T is an infinite c.e. transversal of R and f is a computable function enumerating T, then $\mathrm{Id} \leq R$ via f. Similarly, a ceer is dark if and only if it does not admit any infinite c.e. transversal, that is if and only if all its infinite transversal are immune.

Definition 1.2.3. Given two ceers R and S, their *uniform join* is the ceer $R \oplus S$ defined as follows:

$$x \; R \oplus S \; y \;\Leftrightarrow\; (x = 2u \;\&\; y = 2v \;\&\; u \; R \; v) \text{ or } (x = 2u{+}1 \;\&\; y = 2v{+}1 \;\&\; u \; S \; v),$$

for all x, y.

Clearly, $R \oplus S$ is an upper bound (with respect to $\leq$) of both R and S. The uniform join of finite ceers is finite, as for every $k, h \in \mathbb{N}$, we obviously get $\mathrm{Id}_k \oplus \mathrm{Id}_h \equiv \mathrm{Id}_{k+h}$. Moreover, it is easy to see that dark ceers are closed under $\oplus$. Indeed, let $R \oplus S$ be light, as witnessed by the infinite c.e. transversal T: then T must contain infinitely many even elements or infinitely many odd ones, which form an infinite c.e. transversal of, respectively, R or S. Hence, at least one among R and S must be light.

We have already seen that examples of dark ceers are given by one-dimensional ceers R_X where X is a simple set. Indeed, those are the only dark one-dimensional ceers, as if X is not simple, then $\overline{X}$ contains an infinite c.e. set, meaning that R_X admits an infinite c.e. transversal and hence is light. We conclude this section with the construction of a dark ceer having only finite equivalence classes.

Remark 1.2.4. Throughout the chapter, when we build some ceer R in stages, we usually mean that we define in stages a *computable approximation* of R, namely a sequence of uniformly computable ceers $\{R_s\}_{s\in\mathbb{N}}$, such that $R_0 = \mathrm{Id}$, $R_s \subseteq R_{s+1}$ (in fact, R_{s+1} is obtained by collapsing at most a finite number of R_s-equivalence classes, meaning that all but finitely many R_{s+1}-equivalence classes are indeed singletons), and $R = \bigcup_{s\in\mathbb{N}} R_s$. Furthermore, for every s, we assume to know, through its canonical index, the finite set of elements whose R_s-equivalence classes are not singletons. Finally, when in the construction we say that we R-*collapse* sets $X_1, \dots, X_n$ at some stage s, we mean that we let R_{s+1} be the equivalence relation generated by the sets of pairs $R_s \cup \{(x, y) :\ x, y \in \bigcup_{i\leq n} X_i\}$, namely that we put all elements of $X_1, \dots, X_n$ in a single R-equivalence class.

Proposition 1.2.5. *There exists a dark ceer R such that every R-equivalence class is finite.*

Proof. We want to build a ceer R satisfying the following requirements:

$$\mathcal{R}_e : \text{if } W_e \text{ is infinite, then it is not a transversal of } R,$$

where $(W_e)_{e\in\mathbb{N}}$ is a suitable listing of all c.e. sets, and

$$\mathcal{S}_e : [e]_R \text{ is finite.}$$

The requirements of the first kind, namely requirements $\mathcal{R}_0, \mathcal{R}_1, \dots,$ guarantee that no c.e. set can be a transversal of R, meaning that R is dark. To satisfy some requirement $\mathcal{R}_e$ it is enough to R-collapse, at some stage, two distinct elements $x, y \in W_e$: by this single action, W_e is prevented to be a transversal of R, so that requirement $\mathcal{R}_e$ is permanently satisfied. To satisfy each requirement $\mathcal{S}$, we ensure to R-collapse two different elements only to satisfy some requirement $\mathcal{R}_e$, and we simply order those requirements according to the priority ordering $\mathcal{R}_0 > \mathcal{R}_1, \dots$, so that, if we R-collapse two elements for sake of some requirement $\mathcal{R}_e$, then their R-equivalence classes can be further enlarged only by the finitely many requirements $\mathcal{R}_0, \dots, \mathcal{R}_{e-1}$.

Construction. We build such a ceer R in stages, starting with $R_0 = \mathrm{Id}$. Together with R we enumerate, for each e, a set B_e containing, intuitively speaking, the elements which have been R-collapsed for sake of requirement $\mathcal{R}_e$. For each e, we start with $B_{e,0} = \emptyset$. If B_e is not explicitely redefined at stage $s+1$, then it is understood that $B_{e,s+1} = B_{e,s}$.

At stage $s+1$, let $\mathcal{R}_e$ be the highest priority $\mathcal{R}$-requirement for which $B_{e,s} = \emptyset$ and there are $x \neq y$, $x, y \in W_e \smallsetminus \left(\bigcup_{i<e} B_{i,s}\right)$ with $x \not{R}_s y$, if any, then take such a pair x, y having least pseudocode and R-collapse the equivalence classes of x and y, namely let $B_{e,s} = [x]_{R_s} \cup [y]_{R_s}$ and finally let R_{s+1} be the equivalence relation generated by the set of pairs $R_s \cup \{(x,y) : x, y \in B_{e,s}\}$. In this case, we say that $\mathcal{R}_e$ *acts* at stage $s+1$. Next, we go to the next stage.

Verification. It is clear that each requirement $\mathcal{R}_e$ acts at most once, as if it acts at stage s_0, then for all $s \geq s_0$ we clearly have $B_{e,s} \neq \emptyset$, so that $\mathcal{R}_e$ is never allowed to act at any later stage. In particular, for each e, $B_e = \lim_{s\to\infty} B_{e,s}$ is well-defined, as either $B_e = B_{e,s}$ if $\mathcal{R}_e$ acts at some stage s, or $B_e = \emptyset$ otherwise.

We first claim that R is dark. Indeed, we have already noticed that each requirement $\mathcal{R}_e$ acts at most once, and that whenever $\mathcal{R}_e$ is allowed to act, then this single action is enough to prevent W_e of being a transversal of R, so that $\mathcal{R}_e$ is permanently satisfied. It remains to show that, if W_e is infinite, then $\mathcal{R}_e$ eventually acts. Assume that all requirements $\mathcal{R}_0, \dots, \mathcal{R}_{e-1}$ never require attention after some stage s_0 and that W_e is infinite. If $\mathcal{R}_e$ has already acted at some stage $s \leq s_0$, then there is nothing to prove. Otherwise, as $\bigcup_{i<e} B_{i,s_0} = \bigcup_{i<e} B_i$ is a finite set, while W_e is infinite, there must be a stage $s > s_0$ such that suitable elements x, y are enumerated into W_e, hence $\mathcal{R}_e$ is the current higher priority requirement and therefore is allowed to act.

Finally, we show that each R-equivalence class is finite. We say that $x \neq e$ *injuries* requirement $\mathcal{S}_e$ at stage s whenever $x \mathrel{R_s} e$. Then, it suffices to show that each such requirement is only injured by finitely many elements. Indeed, either e is never R-collapsed to any other number, meaning that $[e]_R$ is a singleton, or there is a highest priority requirement $\mathcal{R}_e$ that, at some stage, R-collapse e to some other number. Hence, $e \in B_e$, so that no other requirement $j > e$ can further enlarge $[e]_R$. But then $\mathcal{S}_e$ is injured at most by the finitely many elements of $\bigcup_{i<e} B_i$.

This concludes the verification and hence the proof. $\square$

1.3 Isomorphisms and strong isomorphisms of ceers

Definition 1.1.1 leads to identify two equivalence relations if they belong to the same degree with respect to computable reducibility, as this means that they are "equally complex". In this chapter we consider two additional ways of identifying ceers, both based on the notion of "isomorphism".

Definition 1.3.1. Given ceers R, S, we say that R and S are *isomorphic* (and write $R \simeq S$) if $R \leq S$ via a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $range(f)$ intersects each S-equivalence class. In this case, we say that f *induces an isomorphism* from R to S.

The choice of the name "isomorphism" is justified by Lemma 1.3.2 below, which has been first proven in [5]. Indeed, following the category theoretic approach to numberings proposed by Ershov [40], equivalence relations on $\mathbb{N}$ can be structured as objects of a category (which will be widely investigated in Section 2). The lemma shows in fact that, when restricting attention only to equivalence relations that are ceers, two objects are isomorphic in the category theoretic sense if and only if they are isomorphic in the sense of our Definition 1.3.1.

Lemma 1.3.2 (Inversion Lemma)**.** *Let R, S be ceers. Then a computable function f induces an isomorphism from R to S if and only if $S \leq R$ via a computable function g such that $g(f(x)) \mathrel{R} x$, and $f(g(x)) \mathrel{S} x$, for all $x \in \mathbb{N}$.*

Proof. Let f induce an isomorphism from R to S. For every x, let $g(x)$ be the first y for which we see that $f(y) \mathrel{S} x$. Then $S \leq R$ via g. Indeed, let $g(u) = x$ and $g(v) = y$: then $u \mathrel{S} f(x) \mathrel{S} f(y) \mathrel{S} v \Leftrightarrow g(u) \mathrel{R} g(v)$, by definition of g and because $R \leq S$ via f. Moreover, by definition of g it is clear that $g(f(x)) \mathrel{R} x$, and $f(g(x)) \mathrel{S} x$, for all $x \in \mathbb{N}$.

Conversely, assume that g witnesses the reduction $S \leq R$ and satisfies both $g(f(x)) \mathrel{R} x$ and $f(g(x)) \mathrel{S} x$, for all $x \in \mathbb{N}$. Clearly, $f(g(x)) \in \text{range}(f) \cap [x]_S \neq \emptyset$, hence range$(f)$ intersects all S-equivalence classes. Moreover, by the properties of $f \circ g$ and $g \circ f$ and since g is a reduction from S

to R, we get

$$x \, R \, y \;\Leftrightarrow\; g(f(x)) \, R \, g(f(y)) \;\Leftrightarrow\; f(x) \, S \, f(y),$$

meaning that f induce an isomorphism from R to S. □

The isomorphism type of every ceer contains a ceer having only infinite classes.

Fact 1.3.3. *For every ceer R there exists a ceer S having only infinite classes and such that $R \simeq S$.*

Proof. Let $\langle \_, \_ \rangle$ denote the Cantor pairing function, and let $(\_)_0$ be its first projection. Given R, let S be such that

$$x \; S \; y \Leftrightarrow (x)_0 \; R \; (y)_0.$$

Clearly, every S-equivalence class is infinite. As is immediate to see, the computable function $(\_)_0$ induces an isomorphism from S to R, since it provides a reduction whose range intersects all R-equivalence classes. □

An even stronger way to identify ceers is to consider only isomorphisms (in the above sense) induced by computable permutations of $\mathbb{N}$.

Definition 1.3.4. We say that ceers R and S are *strongly isomorphic* if $R \leq S$ via a computable permutation f (denoted by $R \simeq_{\mathrm{s}} S$). In this case, we also say that f *induces a strong isomorphism* from R to S.

In contrast with Fact 1.3.3, it is clear that every ceer R having at least one finite class cannot be strongly isomorphic to any ceer S with only infinite classes. On the other hand, the isomorphism type and the strong isomorphism type of a ceer having only infinite classes coincide.

Fact 1.3.5. *If R, S are ceers such that all R-classes and all S-classes are infinite then*

$$R \simeq_s S \Leftrightarrow R \simeq S.$$

Proof. The nontrivial implication $R \simeq S \Rightarrow R \simeq_{\mathrm{s}} S$ follows by a straightforward back-and-forth argument similar to the one used in the proof of the Myhill Isomorphism Theorem. See for instance [2, Remark 1.2] and [1, Lemma 2.3]. □

It is clear that strong isomorphism between two ceers implies two ceers being isomorphic, which in turn implies that those ceers are bi-reducible to each other.

Example 1.3.6. We have already seen that, for all ceers R, S, we have

$$R \simeq_{\mathrm{s}} S \Rightarrow R \simeq S \Rightarrow R \equiv S.$$

Next, we see that all these implications are proper.

Let R be a ceer having one finite equivalence class: by Lemma 1.3.3, there is a ceer S with only infinite equivalence classes such that $R \simeq S$. However, we have already noticed that $R \not\simeq_{\mathrm{s}} S$, since there is a finite R-equivalence class.

Furthermore, notice that, if R-equivalence classes are uncomputable, while S has at least one computable equivalence class, then $R \not\simeq S$. Indeed, assume that $R \simeq S$ via a reduction f witnessing that $R \leq S$ and range(f) intersects each S-equivalence class: then there must be an x such that $[f(x)]_S$ is computable, which implies that $[x]_R$ must be computable as well. In Section 1.4, we will see some examples of *universal* ceers, namely ceers U such that $R \leq U$, for every ceer R. Obviously, those ceers are all bi-reducible to each other. A very natural example of universal ceer is given by *provable equivalence of* PA $\sim_{\mathsf{PA}}$, defined by

$$x \sim_{\mathsf{PA}} y \Leftrightarrow \mathsf{PA} \vdash g(x) \leftrightarrow g(y),$$

where g^{-1} is an bijective Gödel coding of all PA sentences. Notice that $\sim_{\mathsf{PA}}$ has only undecidable equivalence classes. Clearly, $\sim_{\mathsf{PA}} \oplus \mathrm{Id}_1$ is universal as well, as $\sim_{\mathsf{PA}} \leq \sim_{\mathsf{PA}} \oplus \mathrm{Id}_1$. Hence $\sim_{\mathsf{PA}} \equiv \sim_{\mathsf{PA}} \oplus \mathrm{Id}_1$. On the other hand, $\sim_{\mathsf{PA}} \not\simeq \sim_{\mathsf{PA}} \oplus \mathrm{Id}_1$, as the latter ceer has one computable class. Hence, all implications above are proper.

1.4 Some universal ceers

The degree structure of ceers under computable reducibility has been widely investigated. Much effort has naturally been given in the study of Σ^0_1-complete equivalence relation, meaning those ceers that can reduce any ceer to themselves, which are usually called *universal* in this context.

Definition 1.4.1. A ceer U is universal if, for every ceer R, we have that $R \leq U$.

While in the cases of m-complete or 1-complete c.e. sets (and even pairs of disjoint c.e. sets) there are well-known theorems stating that such sets (respectively, pairs of disjoint c.e. sets) are all compubly isomorphic, there are distinct isomorphism types of universal ceers (indeed, they are infinitely many, as shown in [2]). Clearly, only light ceers can be universal.

The following important class of equivalence relations was first considered in [61].

Definition 1.4.2. An equivalence relation $R \neq \mathrm{Id}_1$ is *precomplete* if there is a computable function $f(e, x)$ such that

$$(\forall e, x)\,[\varphi_e(x)\downarrow \Rightarrow\ \varphi_e(x)\ R\ f(e, x)]\,.$$

Such an f is called a *totalizer* for R. Moreover, we say that $f(e, \cdot\,)$ *makes* φ_e *total modulo* R.

The following result gives a useful characterization of precomplete equivalence relations.

Fact 1.4.3 (Ershov's Fixed Point Theorem)**.** *An equivalence relation R is precomplete if and only if there exists a computable function $\widehat{f}$ such that, for every n,*

$$\varphi_n(\widehat{f}(n))\downarrow \Rightarrow\ \varphi_n(\widehat{f}(n))\ R\ \widehat{f}(n).$$

Proof. Assume that R is precomplete, so that there is a computable function $f(n)$ which makes $\varphi_n(n)$ total modulo R, i.e. for any n we have that

$$\varphi_n(n)\downarrow \Rightarrow\ \varphi_n(n)\ R\ f(n).$$

Let $\varphi_{s(n)} = \varphi_n \circ f$ and define $\widehat{f} = f \circ s$. Assume that $\varphi_n(\widehat{f}(n))\downarrow$. We have that

$$\varphi_n(\widehat{f}(n)) = \varphi_n(f(s(n))) = \varphi_n \circ f \circ s(n) = (\varphi_n \circ f) \circ s(n) = \varphi_{s(n)}(s(n)),$$

and hence $\varphi_n(\widehat{f}(n)) = \varphi_{s(n)}(s(n))\ R\ f(s(n)) = \widehat{f}(n)$.

Conversely, assume that $\widehat{f}$ is a computable function such that, for every n, if $\varphi_n(\widehat{f}(n))\downarrow$, then $\varphi_n(n)\ R\ \widehat{f}(n)$. Moreover, fix a partial computable function φ. Define f so that, for all y, $\varphi_{f(n)}(y) = \varphi(n)$. We claim that $g = \widehat{f} \circ f$ makes φ total modulo R. Indeed, if $\varphi(n)\downarrow$, then

$$\varphi(n) = \varphi_{f(n)}(\widehat{f}(f(n)))\ R\ \widehat{f}(f(n)) = g(n),$$

as required. □

Let us call a *diagonal function* of an equivalence relation R any computable function d such that $d(x) \not R\, x$, for all x. The following is an immediate corollary of Fact 1.4.3.

Corollary 1.4.4. *If an equivalence relation has a diagonal function, then it is not precomplete.*

It has been shown in [11] that all precomplete ceers are universal. Moreover, they are all isomorphic to each other, as shown in [56].

Example 1.4.5. Natural examples of precomplete ceers are considered in [92], namely the ceers of Σ_n *provable equivalence*, i.e. the equivalence relation $\sim_n$ defined by

$$x \sim_n y \Leftrightarrow \mathsf{PA} \vdash g_n(x) \leftrightarrow g_n(y),$$

where g_n^{-1} is a bijective Gödel coding of all Σ_n sentences of PA, where $n \geq 1$.

By Corollary 1.4.4, the PA provable equivalence $\sim_{\mathsf{PA}}$ from Example 1.3.6 cannot be precomplete, (hence, in particular, belongs to a different isomorphism type), since logical negation $\neg$ induces a diagonal function. On the other hand, $\sim_{\mathsf{PA}}$ has the weaker property that any partial computable function φ with finite range can be totalized modulo $\sim_{\mathsf{PA}}$: in fact, we can effectively found some $n \geq$ with $\text{range}(\varphi) \subseteq g_n^{-1}(\Sigma_n)$, so that we can actually totalize modulo $\sim_n$. The following definition, due to Montagna ([69]), generalizes this property.

Definition 1.4.6. An equivalence relation R is *uniformly finitely precomplete* (or simply *u.f.p.*) if there is a computable function $f(D, e, x)$ such that, for every finite set D and every $e, x \in \mathbb{N}$,

$$\varphi_e(x) \in [D]_R \ \Rightarrow \varphi_e(x)\ R\ f(D, e, x),$$

where $[D]_R$ denotes the equivalence relation generated by the finite set D.

In the same paper [69], it has been shown that every u.f.p. ceer is universal. We conclude with a characterization of the strong isomorphism type of $\sim_{\mathsf{PA}}$.

Fact 1.4.7 (Bernardi and Montagna, [10])**.** *$R \simeq_s \sim_{\mathsf{PA}}$ if and only if R is u.f.p. and has a diagonal function.*

Chapter 2

On the category of equivalence relations

In his monograph [40] Ershov introduces ad thoroughly investigates the category of numberings. We recall that a *numbering* is a pair $N = \langle \nu, S \rangle$, where S is a nonempty set and $\nu : \mathbb{N} \to S$ is a surjective function. Numberings are the objects of a category $\mathbb{N}$um, called the *category of numberings*; the *morphisms* from a numbering $N_1 = \langle \nu_1, S_1 \rangle$ to a numbering $N_2 = \langle \nu_2, S_2 \rangle$ are the functions $\mu : S_1 \to S_2$ for which there is a computable function f so that the diagram

$$\begin{tikzcd} \mathbb{N} \arrow[r, "f"] \arrow[d, "\nu_1"'] & \mathbb{N} \arrow[d, "\nu_2"] \\ S_1 \arrow[r, "\mu"'] & S_2 \end{tikzcd}$$

commutes. We say in this case that the computable function f *induces the morphism* μ, and we write $\mu = \mu^{N_1,N_2}(f)$.

Now, numberings are equivalence relations in disguise, see our Theorem 2.2.2 below, where we show that the equivalence relations on the set $\mathbb{N}$ of natural numbers can be structured into a category $\mathbb{E}$q which is equivalent to $\mathbb{N}$um. In this paper, we rephrase in $\mathbb{E}$q some of the observations noticed by Ershov about $\mathbb{N}$um, and we point out some useful new facts about $\mathbb{E}$q, and some of its full sabcategories, such as the category $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ of computably enumerable equivalence relations (these relations are called *ceers*; ceers have been widely investigated in the literature, see e.g. [2, 5]), the category $\mathbb{E}\mathrm{q}(\Pi^0_1)$ of co-computably enumerable equivalence relations (called *coceers*; coceers have received much less attention than ceers, but they have been considered in e.g. [8, 73]), and the category $\mathbb{E}\mathrm{q}(\mathrm{Dark}^*)$ whose objects are the dark ceers and the finite ceers. Although in all these categories the monomorphisms trivially coincide with the injective morphisms, we see that in $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ the epimorphisms coincide with the onto morphisms, but in $\mathbb{E}\mathrm{q}(\Pi^0_1)$ there are epimorphisms which are not onto. We also observe that $\mathbb{E}\mathrm{q}$, $\mathbb{E}\mathrm{q}(\Sigma^0_1)$, and $\mathbb{E}\mathrm{q}(\mathrm{Dark}^*)$ are closed under finite products, binary coproducts, and coequalizers. On the other hand, we give an example which shows that $\mathbb{E}\mathrm{q}(\Pi^0_1)$ is not closed under coequalizers.

Let us conclude this introduction by fixing some notations: we denote by $1_{\mathbb{N}}$ the identity function on $\mathbb{N}$, i.e. $1_{\mathbb{N}}(x) = x$. Moreover, for any equivalence relation R on $\mathbb{N}$, let $\mathbb{N}_{/R}$ denotes the set of equivalence classes into which R partitions $\mathbb{N}$: Finally, by $\nu_R : \mathbb{N} \to \mathbb{N}_{/R}$ be given by $\nu_R(x) = [x]_R$, where $[x]_R$ denotes the R-equivalence class of x.

2.1 Preliminaries on category theory

We recall some terminology of category theory which will be used in the rest of the section. The reader is also referred to [60] for these notions.

Basic definitions and equivalence of categories

We start from the scratch, by recalling what a *category* is.

Definition 2.1.1. A *category* $\mathbb{A}$ consists of:

(i) A collection $\mathrm{ob}(\mathbb{A})$ of *objects*.

(ii) For each $A, B \in \mathrm{ob}(\mathbb{A})$, a collection $\mathbb{A}(A, B)$ of *morphism* from A to B.

(iii) For each $A, B, C \in \mathrm{ob}(\mathbb{A})$, a map

$$\begin{aligned} \mathbb{A}(B, C) \times \mathbb{A}(A, B) &\to \mathbb{A}(A, C) \\ (g, f) &\mapsto g \circ f \end{aligned}$$

called *composition*, and for each $A \in \mathrm{ob}(\mathbb{A})$ a function $1_A \in \mathbb{A}(A, A)$ called the *identity* on A such that the composition is associative and, for each $f \in \mathbb{A}(A, B)$, $f \circ 1_A = f = 1_B \circ f$.

Given a category, we can pick only certain of its objects and morphisms: this gives rise to so-called *subcategories*.

Definition 2.1.2. A *subcategory* $\mathbb{B}$ of a category $\mathbb{A}$ consists of a subclass $\mathrm{ob}(\mathbb{B})$ of $\mathrm{ob}(\mathbb{A})$ and, for each $B, B' \in \mathrm{ob}(\mathbb{B})$, a subclass $\mathbb{B}(B, B')$ of $\mathbb{A}(B, B')$ which is closed under composition and identities. Moreover, $\mathbb{B}$ is said to be a *full* subcategory if $\mathbb{B}(B, B') = \mathbb{A}(B, B')$ for every pair $B, B' \in \mathrm{ob}(\mathbb{B})$.

We need to distinguish some special classes of morphisms.

Definition 2.1.3. The morphism $f : A \to B$ is a *monomorphism* if $g = h$ in every commutative diagram of the form

$$
\begin{tikzcd} A \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & B \arrow[r, "f"] & C \end{tikzcd}
$$

i.e. whenever $f \circ g = f \circ h$.

Dually, $f : C \to A$ is an *epimorphism* if $g = h$ for every commutative diagram of the form

$$
\begin{tikzcd} C \arrow[r, "f"] & A \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & B \end{tikzcd}
$$

i.e. when $g \circ f = h \circ f$.

Finally, $f \in \mathbb{A}(A, B)$ is an *isomorphism* if there is $g \in \mathbb{A}(B, A)$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$. In this case, A and B are said to be *isomorphic*, and we write $A \simeq B$.

Another fundamental notions in category theory is the one of *functors*.

Definition 2.1.4. Let $\mathbb{A}$ and $\mathbb{B}$ be categories. A *functor* $F : \mathbb{A} \to \mathbb{B}$ consists of a map

$$
\begin{aligned} \mathrm{ob}(\mathbb{A}) &\to \mathrm{ob}(\mathbb{B}) \\ A &\mapsto F(A) \end{aligned}
$$

and, for each $A, A' \in \mathrm{ob}(\mathbb{A})$, a function

$$
\begin{aligned} \mathbb{A}(A, A') &\to \mathbb{B}(F(A), F(A')) \\ f &\mapsto F(f) \end{aligned}
$$

such that

(i) $F(f' \circ f) = F(f') \circ F(f)$, for each pairs of composable morphisms f, f'.

(ii) $F(1_A) = 1_{F(A)}$, for each $A \in \mathrm{ob}(\mathbb{A})$.

Notice that, for any category $\mathbb{A}$, one can define the functor $1_{\mathbb{A}}$ such that $1_{\mathbb{A}}(A) = A$ for every object A and $1_{\mathbb{A}}(f) = f$ for every morphism f. Moreover, the definition of composition is extended to functors in the obvious way.

We now want to recall the notion of *equivalence* between categories, since the first result we aim to show is that the categories $\mathbb{E}\mathrm{q}$ and $\mathbb{N}\mathrm{um}$ are equivalent (Theorem 2.2.2 below). To do so, we first introduce a class of maps between functors called *natural trasformations*.

Definition 2.1.5. Let $\mathbb{A}, \mathbb{B}$ be categories and $F, G : \mathbb{A} \to \mathbb{B}$ be functors. A *natural trasformations* $\alpha : F \to G$ is a sequence of morphism $(\alpha_A : F(A) \to F(G))_{A \in \mathbb{A}}$ such that for every $f \in \mathbb{A}(A, A')$ the diagram

$$\begin{tikzcd}
F(A) \arrow[r, "F(f)"] \arrow[d, "\alpha_A"'] & F(A') \arrow[d, "\alpha_{A'}"] \\
G(A) \arrow[r, "G(f)"'] & G(A')
\end{tikzcd}$$

commutes.

Note that we can compose natural trasformations: indeed, given natural trasformations $\alpha : F \to G$, $\beta : G \to H$ between functors $F, G, H : \mathbb{A} \to \mathbb{B}$, it is well defined the composition $\beta \circ \alpha = (\beta_A \circ \alpha_A)_{A \in \mathbb{A}}$. Moreover, there is an identity natural trasformation 1_F for any functor $F : \mathbb{A} \to \mathbb{B}$, namely $1_F = (1_{F(A)})_{A \in \mathbb{A}}$. Thus, for any two categories $\mathbb{A}, \mathbb{B}$, there is a category $\mathbb{B}^{\mathbb{A}}$, called the *functor category* from $\mathbb{A}$ to B, whose objects are the functors from A to B and whose morphisms are natural trasformations between them. Since we have a suitable category, to give a good notion of "isomorphism between functors" it is enough to look at isomorphisms in the appropriate functor category.

Definition 2.1.6. Given two functors $F, G \in \mathrm{ob}(\mathbb{B}^{\mathbb{A}})$, a *natural isomorphism* between them is a natural trasformation $\alpha \in \mathbb{B}^{\mathbb{A}}(F, G)$ which is an isomorphism in $\mathbb{B}^{\mathbb{A}}$. In this case, we say that F and G are *naturally isomorphic*.

It is easy to check that a natural transformation α is a natural isomorphism if and only if α_A is an isomorphism, for every $A \in \mathrm{ob}(\mathbb{A})$.

Finally, natural isomorphisms give rise to a good definition for identifying two categories.

Definition 2.1.7. An *equivalence* between categories $\mathbb{A}$ and $\mathbb{B}$ is given by a pair of functors $F : \mathbb{A} \to \mathbb{B}$ and $G : \mathbb{B} \to \mathbb{A}$ such that $G \circ F \simeq 1_{\mathbb{A}}$ and $F \circ G \simeq 1_{\mathbb{B}}$, where $\simeq$ here denotes natural isomorphism of functors. If such an equivalence exits, we say that $\mathbb{A}$ and $\mathbb{B}$ are *equivalent*.

Products and coproducts

We recall the definitions of products and coproducts in a category $\mathbb{A}$, as we will discuss their existence in $\mathbb{E}\mathrm{q}$ and in some of its subcategories of interest in computability theory in Section 2.2.2 below.

Definition 2.1.8. Let $\mathbb{A}$ be a category and $A, B \in \mathrm{ob}(\mathbb{A})$. A *product* of A and B consists of an object $A \times B$ and morphisms $\pi_A : A \times B \to A$, $\pi_B : A \times B \to B$ (called *projections*) such that for every $X \in \mathrm{ob}(\mathbb{A})$ and every pair of morphisms $f_A : X \to A$, $f_B : X \to B$, there exists a unique morphism $f_A \times f_B : X \to A \times B$ which makes the following diagram commute.

$$\begin{tikzcd} & X \arrow[dl, "f_A"'] \arrow[d, dashed, "f_A \times f_B"] \arrow[dr, "f_B"] & \\ A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B \end{tikzcd}$$

Products do not always exists. However, if two objects A and B do have a product, then it must be unique up to isomorphism: hence, we can talk of *the* product $A \times B$.

The dual notion of a product is called a *coproduct*.

Definition 2.1.9. Given $A, B \in \mathrm{ob}(\mathbb{A})$, their (unique up to isomorphism, whenever it exists) *coproduct* is an object $A \amalg B$ together with morphisms $i_A : A \to A \amalg B$, $i_B : B \to A \amalg B$ such that for every $X \in \mathrm{ob}(\mathbb{A})$ and every pair of morphisms $f_A : X \to A$, $f_B : X \to B$, there exists a unique morphism $f_A \amalg f_B : A \amalg B \to X$ which makes the following diagram commute.

$$\begin{tikzcd} & X & \\ A \arrow[ur, "f_A"] \arrow[r, "i_A"'] & A \amalg B \arrow[u, dashed, "f_A \amalg f_B"] & b \arrow[ul, "f_B"'] \arrow[l, "i_B"] \end{tikzcd}$$

The definitions of products and coproducts of two objects in a category $\mathbb{A}$ generalize in the obvious way to the ones for any family of objects. In particular, we obtain a so-called *terminal* object when we consider an empty product, while a so-called *initial* object corresponds to an empty coproduct. Hence, initial and terminal objects do not always exist in a category, but, if so, they are unique up to isomorphism.

Definition 2.1.10. Let $\mathbb{A}$ be a category. An object $T \in \text{ob}(\mathbb{A})$ is *terminal* if for every $A \in \text{ob}(\mathbb{A})$, there is exactly one morphism $A \to T$. Dually, an object $I \in \text{ob}(\mathbb{A})$ is *initial* if for every $A \in \text{ob}(\mathbb{A})$, there is exactly one morphism $I \to A$.

Equalizers and coequalizers

Together with products and coproducts, equalizers and coequalizers form the "basic bricks" to build all sorts of limits and colimits in a category. Indeed, it is known that, if a category has all products, then it has all limits, while having binary products, terminal object and equalizers is enough to have all finite limits. Naturally, dual results hold for colimits.

Definition 2.1.11. In a given category, we call a *fork* objects and morphisms

$$\begin{tikzcd} A \arrow[r, "f"] & X \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & Y \end{tikzcd}$$

such that $s \circ f = t \circ f$, and a *cofork* objects and morphisms

$$\begin{tikzcd} X \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & Y \arrow[r, "f"] & A \end{tikzcd}$$

such that $f \circ s = f \circ t$.

An *equalizer* of two morphism $s, t : X \to Y$ is an object E together with a morphism $i : E \to X$ such that $\begin{tikzcd} E \arrow[r, "i"] & X \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & Y \end{tikzcd}$ is a fork and, for any fork $A \xrightarrow{f} X \overset{s}{\underset{t}{\rightrightarrows}} Y$, there exists a unique morphism $\overline{f} : A \to E$ such that the diagram

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, dashed, "\overline{f}"'] & X \\ E \arrow[ur, "i"] & \end{tikzcd}$$

commutes.

Dually, a *coequalizer* of $s, t : X \to Y$ is an object C together with a morphism $i : C \to X$ such that $X \overset{s}{\underset{t}{\rightrightarrows}} Y \xrightarrow{i} C$ is a cofork and, for any cofork $X \overset{s}{\underset{t}{\rightrightarrows}} Y \xrightarrow{f} A$, there exists a unique morphism $\overline{f} : C \to A$ such that the diagram

$$\begin{tikzcd} Y \arrow[r, "i"] \arrow[dr, "f"'] & C \arrow[d, dashed, "\overline{f}"] \\ & A \end{tikzcd}$$

commutes.

Equalizers and coequalizers do not always exists, but if a pair of morphisms in a category has an equalizer or a coequalizer, this is unique up to isomorphism.

2.2 The category of equivalence relations on $\mathbb{N}$

Following Ershov's category theoretic approach, equivalence relations on $\mathbb{N}$ can be structures as a category $\mathbb{E}$q.

Recall that, for any equivalence relation R on $\mathbb{N}$, we let $\mathbb{N}_{/R}$ denote the set of equivalence classes into which R partitions $\mathbb{N}$, and we let $\nu_R : \mathbb{N} \to \mathbb{N}_{/R}$ be given by $\nu_R(x) = [x]_R$, where $[x]_R$ denotes the R-equivalence class of x.

Definition 2.2.1. If R, S are equivalence relations on $\mathbb{N}$, we say that a function $f : \mathbb{N} \to \mathbb{N}$ is (R, S)-*equivalence preserving* if

$$(\forall x, y)\,[x \; R \; y \;\Rightarrow\; f(x) \; S \; f(y)]\,.$$

Each such f induces a well-defined map $\alpha^{R,S}(f) : \mathbb{N}_{/R} \to \mathbb{N}_{/S}$ given by $[x]_R \mapsto [f(x)]_S$, namely the unique map α such that the following diagram commutes.

$$\begin{tikzcd}
\mathbb{N} \arrow[r, "f"] \arrow[d, "\nu_R"'] & \mathbb{N} \arrow[d, "\nu_S"] \\
R \arrow[r, "\alpha"'] & S
\end{tikzcd}$$

The category $\mathbb{E}\mathrm{q}$ is then defined as follows: its objects are the equivalence relations on $\mathbb{N}$, and, for every pair of equivalence relations R, S, the morphisms $\mathbb{E}\mathrm{q}(R, S)$ consist of all maps $\alpha : \mathbb{N}_{/R} \to \mathbb{N}_{/S}$ such that $\alpha = \alpha^{R,S}(f)$ for some (R, S)-equivalence preserving computable function f.

$\mathbb{E}\mathrm{q}$ is indeed equivalent (in the sense of Definition 2.1.7) to Ershov's category $\mathbb{N}\mathrm{um}$ of numberings.

Theorem 2.2.2. $\mathbb{E}\mathrm{q}$ *and* $\mathbb{N}\mathrm{um}$ *are equivalent categories.*

Proof. We need to exhibit functors $F : \mathbb{N}\mathrm{um} \to \mathbb{E}\mathrm{q}$ and $G : \mathbb{E}\mathrm{q} \to \mathbb{N}\mathrm{um}$ such that $1_{\mathbb{E}\mathrm{q}} \simeq F \circ G$ and $1_{\mathbb{N}\mathrm{um}} \simeq G \circ F$, where here $\simeq$ denotes natural isomorphism between functors. We first define F as follows:

- For any numbering $N = \langle \nu, S \rangle$, let $F(N)$ be the equivalence relation given by $x\ F(N)\ y$ if and only if $\nu(x) = \nu(y)$, for any x, y.
- For any morphism $\mu : N_1 \to N_2$ in $\mathbb{N}\mathrm{um}$, if f is a computable function with $\mu = \mu^{N_1,N_2}(f)$, then let $F(\mu) = \alpha^{F(N_1),F(N_2)}(f)$.

Moreover, let G be defined as follows:

- If R is an equivalence relation, let $G(R) = \langle \nu_R, \mathbb{N}_{/R} \rangle$, with $\nu_R(x) = [x]_R$.
- If $\alpha = \alpha^{R_1,R_2}(f)$ is a morphism in $\mathbb{E}\mathrm{q}$, let $G(\alpha) = \mu^{G(R_1),G(R_2)}(f)$.

Then we have natural trasformations $(\langle N, \nu \rangle \to \langle G(F(N)), \nu_{G(F(N))} \rangle)_{\langle N,\nu \rangle \in \mathbb{N}\mathrm{um}}$ given by $\nu(x) \mapsto \nu_{G(F(N))}(x)$ and $(R \to F(G(R)))_{R \in \mathbb{E}\mathrm{q}}$ given by $[x]_R \mapsto [\nu_R(x)]_{F(G(R))}$, which provide $1_{\mathbb{N}\mathrm{um}} \simeq G \circ F$ and $1_{\mathbb{E}\mathrm{q}} \simeq F \circ G$, as it can easily be checked from the definitions. $\square$

2.2.1 Monomorphisms and epimorphisms of equivalence relations, ceers and coceers

We have recalled the notions of monomorphism and epimorphism in Definition 2.1.3 above. First, we observe that, in $\mathbb{E}\mathrm{q}$, the monomorphisms coincide with the injective morphisms.

Proposition 2.2.3. *In* $\mathbb{E}\mathrm{q}$*, the the monomorphisms coincide with the injective morphisms*

Proof. It is trivial to check that every injective morphism in $\mathbb{E}\mathrm{q}$ is a monomorphism.

Conversely, let $\gamma : R \to S$ be a monomorphism induced by some computable function f and assume that γ is not injective. Then there must be distinct equivalence classes $[a_1]_R \neq [a_2]_R$ with $\gamma([a_1]_R) = \gamma([a_2]_R)$. For $i = 1, 2$, define constant (hence clearly computable) functions $g_i(x) = a_i$. Then, for every equivalence relation E, the functions g_1 and g_2 induce distinct morphisms $\alpha_1 = \alpha^{E,R}(g_1), \alpha_2 = \alpha^{E,R}(g_2) : E \to R$, such that $\gamma \circ \alpha_1 = \gamma \circ \alpha_2$, showing that γ is not a monomorphism. $\square$

Remark 2.2.4. Given equivalence relation R, S, it is easy to see that $R \leq S$, as defined in Definition 1.1.1, if and only if there exists an injective morphism $\mu : R \to S$. Hence, from the point of view of category theory, $R \leq S$ may also be expressed by saying that R is a *subobject* of S, see MacLane [60, p. 122] and Ershov [39, 40].

Another easy observation is that any surjective morphism in $\mathbb{E}\mathrm{q}$ is, in fact, an epimorphism. Indeed, let $\gamma : R \to S$ be a surjective morphism and assume that α and β are morphisms with the property that $\alpha \circ \gamma = \beta \circ \gamma$. If $\alpha \neq \beta$, then there must be some x such that $\alpha([x]_S) \neq \beta([x]_S)$. But since γ is surjective, there is some y such that $[x]_S = \gamma([y]_R)$, meaning that $\alpha(\gamma([y]_S)) \neq \beta(\gamma([y]_S))$, contradicting our assumption.

However, the converse implication is not always true.

Theorem 2.2.5. *In* $\mathbb{E}\mathrm{q}$*, there are epimorphisms which are not surjective.*

Proof. Let A, B be two disjoint undecidable Π^0_1 sets such that their union is undecidable. For instance take $A = 2\overline{K}$ and $B = 2\overline{K} + 1$, where $\overline{K}$ denotes any uncomputable co-c.e. set, so that $A \cup B = \overline{K} \oplus \overline{K}$ is still uncomputable and Π^0_1. Consider the coceer R whose equivalence classes are A, B and then all singletons: i.e. $x \mathrel{R} y$ if and only if $x = y$ or $x, y \in A$ or $x, y \in B$.

Since $C = \overline{A \cup B}$ is an infinite c.e. set, we can fix a computable bijection $f : \mathbb{N} \to C$, which clearly provides a reduction $f : \mathrm{Id} \leq R$, such that the range of f is C. Then the monomorphism $\alpha = \alpha^{\mathrm{Id},R}(f)$ is not surjective, as $A, B \nsubseteq \mathrm{range}(f)$. We claim that α is an epimorphism. To prove this claim, let $\alpha_1 = \alpha^{R,S}(f_1)$, $\alpha_2 = \alpha^{R,S}(f_2)$, for some coceer S and computable functions f_1, f_2, be distinct morphisms such that $\alpha_1 \circ \alpha = \alpha_2 \circ \alpha$. Given the latter condition, these morphisms must be equal on C and may be distinct only because of the values they take on A and B. Therefore, we distinguish the following cases:

- $\alpha_1(A) \neq \alpha_2(A)$ and $\alpha_1(B) = \alpha_2(B)$. Then

  $$(\forall x)[x \in \overline{A} \Leftrightarrow f_1(x) \; S \; f_2(x)],$$

  giving that $\overline{A}$ is co-c.e., hence A is computable, contradiction.

- $\alpha_1(A) = \alpha_2(A)$ and $\alpha_1(B) \neq \alpha_2(B)$. A similar argument as in the previous item shows that B is computable, contradiction.

- $\alpha_1(A) \neq \alpha_2(A)$ and $\alpha_1(B) \neq \alpha_2(B)$. In this case

  $$(\forall x)[x \in \overline{A \cup B} \Leftrightarrow f_1(x) \; S \; f_2(x)],$$

  showing that $A \cup B$ is computable, which is again a contradiction.

Thus, α gives an example of epimorphism which is not surjective, as claimed. $\square$

It is then natural to restrict our attention to full subcategories of $\mathbb{E}\mathrm{q}$ (as in Definition 2.1.2) of major interest in computability theory, namely those of ceers and coceers, which are denoted, respectively, by $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ and $\mathbb{E}\mathrm{q}(\Pi^0_1)$. Analogue notation is used for the full subcategory whose objects are the equivalence relations sitting on a certain level of the arithmetical hierarchy.

In the proof of Theorem 2.2.5, we provide a co-ceer R and an epimorphism $\alpha : \mathrm{Id} \to R$ which is not surjective. Therefore, we immediately get the following corollary.

Corollary 2.2.6. *In $\mathbb{E}\mathrm{q}(\Pi^0_1)$ there are epimorphisms which are not surjective.*

On the other hand, the coincidence between epimorphisms and surjective morphisms holds in the realm of ceers.

Theorem 2.2.7. *In $\mathbb{E}\mathrm{q}(\Sigma_1^0)$ the epimorphisms coincide with the surjective morphisms.*

Proof. Let R, S be ceers and consider a morphism $\alpha : R \to S$ which is induced by some computable function f. Assume that α is not surjective, so that there exists an element a with $[a]_S \notin \mathrm{range}(\alpha)$. Moreover, let

$$A = \{x : \ (\exists y)[y \in \mathrm{range}(h) \text{ and } x \ S \ y]\}.$$

Finally, let T be any precomplete ceer and f be a totalizer of T (as defined in Definition 1.4.2).

We are going to define two (S, T)-equivalence preserving computable functions g_1, g_2 inducing distinct morphisms $\alpha_1 = \alpha^{S,T}(g_1)$ and $\alpha_2 = \alpha^{S,T}(g_2)$ which coincide (in fact, are costant) on range(α), thus witnessing that α cannot be an epimorphism. Our construction is somewhat modelled on the proofs of [85, Theorem 2.6 and Corollary 2.8].

It is easy to check that any precomplete ceer has infinitely many equivalence classes: hence, let b, c_1, c_2 be such that their T-equivalence classes are pairwise disjoint. Without loss of generality, we assume to work with computable approximations $(A_s)_{s\in\mathbb{N}}$ of A, $(S_s)_{s\in\mathbb{N}}$ of S and $(T_s)_{s\in\mathbb{N}}$ (see Remark 1.2.4) such that

(i) for every s, $A_{2s+2} = A_{2s+1}$ and $S_{2s+2} = S_{2s+1}$; and

(ii) if $i \notin A_s$ and $i \not\mathrel{S_s} a$ and $j \ S_s \ a$, then $j \notin A_s$.

We give the construction of g_1 below: the construction of g_2 is analogous, but interchanging c_1 and c_2 at each stage.

Construction. First notice that, by property (i), neither S nor A change at even stages, which are then devoted to ensure that $[c_2]_T$ has not be placed in the range of α_1, and that g_1 is (S, T)-equivalence preserving. We let $g_1(i) = f(e, i)$, where e is an index of a partial computable function, which we control by the Recursion Theorem. Therefore, we actually define φ_e in stages. At any stage, we may call a special clause $(\star)$ which, if called, "freezes" the construction. At stage $s + 1$, if clause $(\star)$ has not be called at any previous stage, we want to satisfy the following inductive assumption:

If $i \notin A_s$, $i \not\mathrel{S_s} a$ and $\varphi_{e,s}(i)\downarrow$, then i is not least in its equivalence class. (†)

The construction then goes as follows.

Stage 0. Let $\varphi_{e,0}(i)\uparrow$ for all i.

Stage $2s+1$. If we have called $(\star)$ at any previous stage, let $\varphi_{e,s+1} = \varphi_{e,s}$. Otherwise, if $\varphi_{e,s}(i)\uparrow$, then:

1. (a) if $i \in A_s$, let $\varphi_{e,s+1}(i) = b$;

   (b) otherwise, if $i \; S_s \; a$, let $\varphi_{e,s+1}(i) = c_1$;

2. otherwise, if $i \notin A_s$, $i \not S_s \; a$ and there exists $j < i$ such that $j \; S_s \; i$ and $\varphi_{e,s}(j) \uparrow$ (thus, by property (ii), $j \notin A - s$), let $\varphi_{e,s+1}(i) = f(e, j)$ for the least such j.

It is straightforward to check that, in both cases, our inductive assumption (†) is preserved.

Stage $2s+2$. If we have called ($\star$) at any previous stage, let $\varphi_{e,s+1} = \varphi_{e,s}$. Otherwise, if $i \neq A_s$. $i \not S_s \; a$ and $\varphi_{e,s}(i) \uparrow$, then do the following:

1. if $f(e,i) \; T_s \; b$ or $f(e,i) \; T_s \; c_1$, then let $\varphi_{e,s+1}(i) = c_2$, and call clause ($\star$);

2. otherwise, if $f(e,i) \; T_s \; c_2$, then define $\varphi_{e,s+1}(i) = c_1$, and call clause ($\star$).

Notice that, by (†), if i is such that $i \notin A_s$ and $i \not S_s \; a$, there is some $j < i$ with $j \; S_s \; i$, $j \notin A_s$ and $i \not S_s \; a$, so that we certainly act on j at some odd stage.

Verification. The verification is based on the following claims.

Claim 1. *We never call clause ($\star$).*

Proof of claim. Assume that we call clause ($\star$) in case 1. at some even stage. Then there must be some i with $\varphi_e(i) = c_2 \; T \; f(e,i)$, but it also holds either $f(e,i) \; T_s \; b$ or $f(e,i) \; T_s \; c_1$, which contradicts the fact that $[b]_T, [c_1]_T, [c_2]_T$ are pairwise disjoint. Similarly, if we call ($\star$) in case 2. at some even stage, we must have some i with $\varphi_e(i) = c_1 \; T \; f(e,i)$, but also $f(e,i) \; T \; c_2$, which implies $c_1 \; T \; c_2$, another contradiction. ■

Claim 2. *If $\varphi_e(i) \uparrow$, then $i \notin A \cup [a]_S$, and i is the least of its equivalence class.*

Proof of claim. If i is not least of its equivalence relation or $i \in A \cup [a]_S$, it is clear from the construction that we must define $\varphi_e(i)$ at some odd stage. ■

Claim 3. *If $i \in A$ then $f(e,i) \; T \; b$, and if $i \in [a]_S$ then $f(e,i) \; T \; c_1$.*

Proof of claim. By Claim 2, $\varphi_e(i)$ is defined and there are $i_0 < i_1 < \cdots < i_k = i$ which satisfies $i_h \; S \; i_k$ for every $h, k \leq n$, and $\varphi_e(i_k) = f(e, i_{k-1})$ for every $0 < k \leq n$ (through case 2. at some odd stage), whereas $\varphi_e(i) = b$ in case $i_0 \in A$ (as case 1.(a) applies), or $\varphi_e(i) = c_1$ if $i_0 \in [a]_S$ (so that case 1.(b) applies). Since T is precomplete via the totalizer f, we have that

$$\varphi_e(i_0)\ T\ f(e,i_0) = \varphi_e(i_1)\ T\ f(e,i) = \varphi_e(i_2)\ T\ \ldots \cdots T\ f(e,i_n) = f(e,i).$$

Hence, we get that $f(e,i)\ T\ b$ in case $i \in A$, or $f(e,i)\ T\ c_1$ if $i\ S\ a$. ■

Claim 4. *If $i, j \notin A \cup [a]_S$ and $i\ S\ j$ then $f(e,i)\ T\ f(e,j)$.*

Proof of claim. Assume that $[i]_S = [j]_S = \{i_0 < i_1 < \ldots\}$. Then $\phi_e(i_0)$ is undefined, and by induction on n it is easy to see (by an argument similar to the previous item, since if $h > 0$ then $\phi_e(i_h)$ is defined through case 2. of an odd stage), that $f(e,i_h)\ T\ f(e,i_0)$. ■

Claim 3 and Claim 4 together show that g_1 is (S,T)-equivalence preserving, meaning that α_1 is well-defined. Moreover, by Claim 1, $[c_2]_T \notin \text{range}(\alpha_1)$ (as we never acted through 1. at even stages). On the other hand, $\alpha_1([a]_S) = [c_1]_T$.

In a similar way, but interchanging c_1 and c_2 at each stage, we define a computable function g_2 which induces a morphism α_2 with the properties that $[c_1]_T \notin \text{range}(\alpha_2)$ and $\alpha_2([a]_S) = [c_2]_T$. We hence get that $\alpha_1 \circ \alpha = \alpha_2 \circ \alpha$, as $\alpha_1([x]_S) = \alpha_2([x]_S) = b$ for every $[x]_S \in \text{range}(\alpha)$, but $\alpha_1 \neq \alpha_2$, because $[c_1]_T \in \text{range}(\alpha_1) \smallsetminus \text{range}(\alpha_2)$ and $[c_2]_T \in \text{range}(\alpha_2) \smallsetminus \text{range}(\alpha_1)$. Thus, α cannot be an epimorphism. □

2.2.2 Products and coproducts in $\mathbb{E}\text{q}$

This section is devoted to investigate the existence of products and coproducts (see Definitions 2.1.8 and 2.1.9) in the category of equivalence relations. The following simple fact was already observed in [40].

Theorem 2.2.8. $\mathbb{E}\text{q}$ *has all nonempty finite products and nonempty finite coproducts.*

Proof. Given equivalence relations R, S, consider the triple $(R \times S, \pi_R, \pi_S)$, where

$$\langle x, y\rangle\ R \times S\ \langle u, v\rangle \iff x\ R\ u \text{ and } y\ S\ v,$$

with $\pi_R = \alpha^{R\times S,R}(p_0)$ and $\pi_S = \alpha^{R\times S,S}(p_1)$ induced, respectively by the first and second projections of the Cantor pairing function $\langle -,-\rangle$. Let T be an equivalence relation and consider morphisms $\rho_R : T \to R$ and $\rho_S : T \to S$

induced, respectively, by computable functions f_R and f_S: to show that $(R \times S, \pi_R, \pi_S)$ is a product in the category theoretical sense, we first need to find a morphism $\rho_R \times \rho_S$ which makes the following diagram commute.

$$\begin{tikzcd}
 & T \arrow[dl, "\rho_R"'] \arrow[d, dashed, "\rho_R \times \rho_S"] \arrow[dr, "\rho_S"] & \\
R & R \times S \arrow[l, "\pi_R"] \arrow[r, "\pi_S"'] & S
\end{tikzcd}$$

But for this it is enough to take $\rho_R \times \rho_S = \alpha^{T, R\times S}(f_R \times f_S)$, where $f_R \times f_S(x) = \langle f_R(x), f_S(x)\rangle$. To show uniqueness, suppose that $\beta : T \to R \times S$ makes the defining diagram commute. Then if $\beta([x]_T) = [\langle u, v\rangle]_{R\times S}$ we have that $\pi_R([\langle u, v\rangle]_{R\times S}) = [u]_R = \rho_R([x]_T) = [f_R(x)]_R$, and thus $u \ R \ f_R(x)$, and similarly $v \ S \ f_S(x)$, giving $\langle u, v\rangle \ R \times S \ \langle f_R(x), f_S(x)\rangle$. This yields

$$\rho_R \times \rho_S([x]_T) = [\langle f_R(x), f_S(x)\rangle]_{R\times S} = [\langle u, v\rangle]_{R\times S} = \beta([x]_T),$$

i.e., $\beta = \rho_R \times \rho_S$.

Now, we show that the coproduct of equivalence relations R, S corresponds to their uniform join (already introduced in Definition 1.2.3), i.e. the equivalence relation $R \oplus S$ generated by the set of pairs

$$\{(2x, 2y) : \ x \ R \ y\} \cup \{(2x+1, 2y+1) : \ x \ S \ y\},$$

together with morphisms $i_R = \alpha^{R, R\oplus S}(x \mapsto 2x)$, $i_S : \alpha^{S, R\oplus S}(x \mapsto 2x+1)$. Again, given an equivalence relation T together with morphisms $\rho_R : T \to R$ and $\rho_S : T \to S$ induced, respectively, by computable functions f_R and f_S, we need to show a morphism $\rho_R \amalg \rho_S$ making the following diagram commute.

$$\begin{tikzcd}
 & T & \\
R \arrow[ur, "\rho_R"] \arrow[r, "i_R"'] & R \oplus S \arrow[u, dashed, "\rho_R \amalg \rho_S"] & S \arrow[ul, "\rho_S"'] \arrow[l, "i_S"]
\end{tikzcd}$$

Then we take $\rho_R \amalg \rho_S = \alpha^{R\oplus S, T}(f_R \oplus f_S)$, where

$$f_R \oplus f_S(x) = \begin{cases} f_R(y), & \text{if } x = 2y, \\ f_S(y), & \text{if } x = 2y+1. \end{cases}$$

To show uniqueness, suppose that $\beta : R \oplus S \to T$ makes the defining diagram commute: then

$$\beta([2x]_{R\oplus S}) = \beta(i_R([x]_R)) = \rho_R([x]_R) = (\rho_R \amalg \rho_S)([2x]_{R\oplus S}),$$

and similarly $\beta([2x+1]_{R\oplus S}) = (\rho_R \amalg \rho_S)([2x+1]_{R\oplus S})$. □

Clearly, if both R and S are Σ^0_n (respectively, Π^0_n) equivalence relations, then also $R \times S$ and $R \oplus S$ are are Σ^0_n (Π^0_n). This gives immediately the following corollary.

Corollary 2.2.9. *For every n, $\mathbb{E}\mathrm{q}(\Sigma^0_n)$ and $\mathbb{E}\mathrm{q}(\Pi^0_n)$ have all nonempty finite products and coproducts.*

We now turn to terminal and initial object in $\mathbb{E}\mathrm{q}$, which corresponds, respectively, to the empty product and the empty coproduct, which we have presented in Definition 2.1.10.

Theorem 2.2.10. $\mathbb{E}\mathrm{q}$ *has a terminal object (and hence it has all finite products), but has no inital object.*

Proof. Recall that a terminal object in $\mathbb{E}\mathrm{q}$ would be an equivalence relation T such that for every equivalence relation R there exists a unique morphism $R \to T$. But then it is easy to see that Id_1 is indeed a terminal object. Dually, an initial object in $\mathbb{E}\mathrm{q}$ is defined as an equivalence relation I such that for any equivalence relation R there is a unique morphism $I \to R$: thus, no equivalence relation X can be initial, as there are always two distinct morphisms $X \to \mathrm{Id}_2$. □

Since Id_1 is computable, we immediately get the following corollary.

Corollary 2.2.11. *For every n, $\mathbb{E}\mathrm{q}(\Sigma^0_n)$ and $\mathbb{E}\mathrm{q}(\Pi^0_n)$ have a terminal object (hence, all finite products) but no initial object.*

2.2.3 Equalizers and coequalizers

The terminology used in this section has been presented in Definition 2.1.11.

First, it is easy to see that equalizers do not always exists in $\mathbb{E}\mathrm{q}$. For example, consider the computable functions f_0 and f_1 defined by $f_0(x) = 0$ and $f_1(x) = 1$, for every x, and let $\alpha_0 = \alpha^{\mathrm{Id},\mathrm{Id}}(f_0)$, $\alpha_1 = \alpha^{\mathrm{Id},\mathrm{Id}}(f_1)$. Let $\beta : R \to \mathrm{Id}$ be a morphism induced by some computable function f: then

$R \xrightarrow{\beta} \mathrm{Id} \underset{\alpha_1}{\overset{\alpha_0}{\rightrightarrows}} \mathrm{Id}$ cannot be a fork, since $\alpha_0(\beta([x]_R)) = \alpha_0([f(x)]_{\mathrm{Id}}) = [f_0(f(x))]_{\mathrm{Id}} = 0$, but $\alpha_1(\beta([x]_R)) = \alpha_1([f(x)]_{\mathrm{Id}}) = [f_1(f(x))]_{\mathrm{Id}} = 1$, hence $\alpha_0 \circ \beta \neq \alpha_1 \circ \beta$.

The situation of coequalizers is much more interesting.

Theorem 2.2.12. $\mathbb{E}\mathrm{q}$ *has all coequalizers.*

Proof. Let X, Y be equivalence relations, and $\alpha, \beta : X \to Y$ be morphisms, with $\alpha^{X,Y} = \alpha(f_1)$ and $\beta = \alpha^{X,Y}(f_2)$, for suitable computable functions f_1, f_2. Consider the equivalence relation Z generated by the set of pairs $Y \cup \{(f_1(x), f_2(x)) : x \in \mathbb{N}\}$ and the surjective morphism $\gamma : Y \to Z$ induced by $1_{\mathbb{N}}$. We claim that (Z, γ) is the coequalizer of α and β. First, notice that $X \underset{\beta}{\overset{\alpha}{\rightrightarrows}} Y \xrightarrow{\gamma} Z$ is a cofork. Indeed, for any x we have $f_1(x)\ Z\ f_2(x)$, hence

$$\gamma(\alpha([x]_X)) = \gamma([f_1(x)]_Y) = [f_1(x)]_Z = [f_2(x)]_Z = \gamma([f_2(x)]_Y) = \gamma(\beta([x]_X)).$$

Consider an equivalence relation U and a computable function g such that $\alpha^{Y,U}(g) \circ \alpha = \alpha^{Y,U}(g) \circ \beta$. It remains to check that $\alpha^{Y,U}(g) = \alpha^{Z,U} \circ \gamma$, but this is obvious from the fact that $[x]_Y \subseteq [x]_Z$. $\square$

Notice that, if Y is a Σ^0_n equivalence relation, then the equivalence relation Z defined above (namely, Z generated by $Y \cup \{(f : 1(x), f_2(x)) : x \in \mathbb{N}\}$, for any pair of computable functions f_1, f_2) is clearly Σ^0_n as well. Thus, we immediately obtain the following corollary.

Corollary 2.2.13. *For every n, $\mathbb{E}\mathrm{q}(\Sigma^0_n)$ has all coequalizers.*

In particular, every ceer can be seen as the coequalizer of a pair of morphisms $\alpha, \beta : \mathrm{Id} \to \mathrm{Id}$.

Corollary 2.2.14. *Let R be an object of $\mathbb{E}\mathrm{q}(\Sigma^0_1)$. Then there is a morphism γ such that (R, γ) is a coequalizer of morphisms* $\mathrm{Id} \underset{\beta}{\overset{\alpha}{\rightrightarrows}} \mathrm{Id}$.

Proof. Since R is a ceer, there is a computable function h which enumerates R, namely $u\ R\ v$ if and only if $\langle u, v\rangle \in \mathrm{range}(h)$. Let $f_1(x) : p_0(h(x))$ and $f_2(x) = p_1(h(x))$, where p_0 and p_1 denote the projections of Cantor pairing function. But then R is generated by the pairs $\{(f_1(x), f_2(x)) : x \in \mathbb{N}\}$. By the proof of Theorem 2.2.12, R, together with the morphism γ induced by h, is therefore the coequalizer of the pair of morphisms induced by f_1 and f_2. $\square$

Unfortunately, building a coequalizer as in Theorem 2.2.12 do not always produce a Π^0_n equivalence relation Z when starting from Π^0_n equivalence relations X and Y. We end this section by showing that this is the case even for $n = 1$, which is a consequence of the following result.

Lemma 2.2.15. *There exist computable functions f_1 and f_2 and a coceer Y such that the equivalence relation Z generated by the set of pairs $Y \cup \{(f_1(x), f_2(x)) : x \in \mathbb{N}\}$ has exactly two equivalence classes, at least one of which is not Π^0_1, and hence Z is not Π^0_1.*

Proof. We construct in stages a coceer Y and a c.e. set U: at each stage s, we define an equivalence relation Y_s, consisting of finitely many singletons and a co-finite class, and a finite set U_s such that $(Y_s)_{s\in\mathbb{N}}$ is a computable approximation of Y (i.e. $(Y_s)_{s\in\mathbb{N}}$ is a computable sequence with $Y_s \supseteq Y_{s+1}$, and $Y = \bigcap_s Y_s$) and $(U_s)_{s\in\mathbb{N}}$ is a computable approximation of U (namely, U_s is a computable sequence with $U_s \subseteq U_{s+1}$, and $U = \bigcup_s U_s$). We work with computable approximations $\{V_{e,s} : e, s \in \mathbb{N}\}$ to the Π^0_1 sets (meaning that the predicate "$x \in V_{e,s}$" is computable in e, x, s, $V_{e,s} \supseteq V_{e,s+1}$ and $V_e = \bigcap_s V_e, s$, for all e, s). The construction is as follows.

Stage 0. Let Y_0 be the equivalence relation consisting of the two equivalence classes $\{0\}$ and $\mathbb{N} \smallsetminus \{0\}$ (namely, $x \; Y_0 \; y \;\; \Leftrightarrow x = y = 0$ or $(x \neq 0$ and $y \neq 0))$ and $U_0 = \emptyset$.

Stage $s + 1$. For every e, if $e + 2 \in V_{e,s} \smallsetminus V_{e,s+1}$, then we Y-*isolate* $e + 2$ at stage $s + 1$, namely we let Y_{s+1} be the equivalence relation generated by the set of pairs $Y_s \smallsetminus \{(e + 2, y) : e + 2 \in V_{e,s} \smallsetminus V_{e,s+1}, y \neq e + 2\}$, so that $[e + 2]_{Y_{s+1}} = \{e + 2\}$. Moreover, let $U_{s+1} = U_s \cup \{e + 2 \in V_{e,s} \smallsetminus V_{e,s+1}\}$.

This ends the construction. Let $f_1(x) = 0$ for all x and f_2 be any computable function with range$(f_2) = 2$. Finally, let Z be the equivalence relation generated by $Y \cup \{(f_1(x), f_2(x)) : x \in \mathbb{N}\}$, namely by Y together with the set of pairs $\{(0, e+2)\}$ such that $e+2$ has been Y-isolated at some stage.

We now check that the equivalence relation Z so obtained has the required properties. First, clearly $[e+2]_Y \cap ([0]_Y \cup [1]_Y)$ if and only if $e+2 \notin V_e$, which in turn implies $0 \; Z \; e + 2$ if and only if $e + 2 \notin V_e$. Hence, $[0]_Z \neq V_e$ for every e, meaning that $[0]_Z$ is not Π^0_1. On the other hand, for all numbers x distinct from 0 and from those $e + 2$ which have been Y-isolated at some stage, we have $x \; Y \; 1$ (as they were at stage 0), and consequently $x \; Z \; 1$. Thus, Z has exactly two equivalence classes: since $[0]_Z$ is not Π^0_1, Z cannot be Π^0_1. $\square$

Theorem 2.2.16. *There is a coceer Y and morphisms* $\mathrm{Id} \underset{\beta}{\overset{\alpha}{\rightrightarrows}} Y$ *such that their coequalizer in $\mathbb{E}\mathrm{q}$ is properly Σ^0_2. Hence, $\mathbb{E}\mathrm{q}(\Pi^0_1)$ is not closed under coequalizers.*

Proof. Let f_1, f_2, Y, Z be as in Lemma 2.2.15: notice that Z is a Σ^0_2 relations, indeed

$$x \; Z \; y \;\Leftrightarrow\; x = y \text{ or } (x \neq 0 \text{ and } (\exists e \forall s)\, y \in V_{e,s}) \text{ or } (x = 0 \text{ and } (\exists e, s)\, y \notin V_{e,s}).$$

Hence, by Theorem 2.2.12, the identity $1_{\mathbb{N}}$ induce a morphism $\gamma : Y \to Z$ such that (Z, γ) is the coequalizer of $\alpha = \alpha^{\mathrm{Id},Y}(f_1)$ and $\beta = \alpha^{\mathrm{Id},Y}(f_2)$ in the category $\mathbb{E}\mathrm{q}(\Sigma^0_2)$. On the other hand, Lemma 2.2.15 ensures that $Z \notin \Pi^0_1$. $\square$

While we show that $\mathbb{E}\mathrm{q}(\Pi^0_1)$ is not closed under coequalizers, the above observation does not let us conclude that $\mathbb{E}\mathrm{q}(\Pi^0_1)$ does not have coequalizers: we hence conclude this section by raising the following question.

Question 2.2.17. Does $\mathbb{E}\mathrm{q}(\Pi^0_1)$ have coequalizers?

2.2.4 Subcategories of $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ and closure under binary coproducts and coequalizers

A well-known fact in category theory is that a category has all finite colimits if and only if it has coequalizers and finite coproducts (including the empty coproduct, namely an initial object). We have already seen (Theorems 2.2.8 and 2.2.10 above) $\mathbb{E}\mathrm{q}$ has all nonempty finite colimits, but it lacks an initial object: however, binary coproducts and coequalizers can be still used to build all those finite colimits which do not need an initial object.

We have already observed that, for every n, $\mathbb{E}\mathrm{q}(\Sigma^0_n)$ is closed under coequalizers and nonempty finite coproducts, although it does not have an initial object: see Corollaries 2.2.9, 2.2.11 and 2.2.13 above.

Focusing on the full subcategory $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ of ceers, it might be of some interest to know whether the classes of dark ceers and of light ceers, which we have introduced in Definition 1.2.1, allows for the same constructions of finite colimits, namely if the same closure properties holds. Corollary 2.2.14, which shows that every ceer can be seen as the coequalizer of suitable morphisms $\mathrm{Id} \overset{\alpha}{\underset{\beta}{\rightrightarrows}} \mathrm{Id}$, excludes that $\mathbb{E}\mathrm{q}(\mathbf{Light})$ (or even $\mathbb{E}\mathrm{q}(\mathbf{Light} \cup \mathcal{F})$) is closed under coequalizers, since Id is obviously light. $\mathbb{E}\mathrm{q}(\mathbf{Dark})$ is not closed under coequalizers, too, because of the following observation.

Remark 2.2.18. The coequalizers of two dark ceers, as built in Theorem 2.2.12, can be finite. In fact, consider, for instance, the pair of morphisms $X \overset{\alpha}{\underset{\beta}{\rightrightarrows}} X \oplus \mathrm{Id}$, , with α and β being induced, respectively, by the function $x \mapsto 2x$ and $x \mapsto 1$.

2.3 Conclusion

In this chapter we have studied several properties of the category $\mathbb{E}\mathrm{q}$ of equivalence relations: this extends Ershov's investigation of the category $\mathbb{N}\mathrm{um}$ of numberings, as these two categories are proven to be equivalent.

We have first observed that in $\mathbb{E}\mathrm{q}$ the monomorphisms coincide with the injective morphisms. On the other hand, in $\mathbb{E}\mathrm{q}$ there are epimorphisms which are not surjective: in particular, we have seen that epimorphisms and surjective morphisms coincide in $\mathbb{E}\mathrm{q}(\Sigma^0_1)$, but this is no longer the case in $\mathbb{E}\mathrm{q}(\Pi^0_1)$.

Furthermore, we have shown that $\mathbb{E}\mathrm{q}$ has all finite products (including the empty one, namely a terminal object) and all non-empty finite coproducts, but does not have the empty coproduct (i.e. an initial object): moreover, the same results hold when we restrict ourselves to the equivalence relations in each level of the arithmetical hierarchy.

Finally, we have observed that equalizers do not always exists in $\mathbb{E}\mathrm{q}$, but coequalizers do. In particular, we have shown that, for every n, $\mathbb{E}\mathrm{q}(\Sigma^0_n)$ has all coequalizers. However, this closure properties fails if we only consider the subcategories of $\mathbb{E}\mathrm{q}(\Sigma^0_1)$ of, respectively, light and dark ceers. On the other $\mathbb{E}\mathrm{q}(\Pi^0_1)$ is not closed under coequalizers: indeed, the question of whether $\mathbb{E}\mathrm{q}(\Pi^0_1)$ has coequalizers is open.

Chapter 3

The expressive power of algebraic structures by means of computable reducibility

Computably enumerable equivalence relations, or ceers, have been an active field of research in recent years. A great deal of the interest in ceers certainly is due to the fact that they appear quite often in mathematical logic (where they appear, for instance, as the relations of provable equivalence in formal systems), and in general mathematics and computer science where they appear as word problems of effectively presented familiar algebraic structures. An important example in this sense is the word problem for finitely presented (or, f.p.) groups. If $\langle X; R \rangle$ is a f.p. group and one codes the universe of the free group F_X on X with $\mathbb{N}$, then the *word problem* of the group is the ceer that identifies two elements $x, y \in F_X$ if xy^{-1} lies in the normal subgroup of F_X generated by the relators appearing in the relation R of the presentation of the group. The word problem of a f.p. group can be decidable (i.e. the corresponding ceer is decidable), but also undecidable, and in fact can be of any c.e. Turing degree: this was obtained independently by Fridman [42], Clapham [28] and Boone [21, 22, 23] (despite the difference in publication dates, the work of these authors was essentially simultaneous).

Of course not every ceer can be the word problem of a f.p. group, or even of a computably enumerable (c.e.) group, see Definition 3.1.1 below. For instance, the equivalence classes of the word problem of a c.e. group are uniformly computably isomorphic with each other: to show that the equivalence class of u is isomorphic to the equivalence class of v, just use the mapping $x \mapsto xu^{-1}v$. Therefore no ceer having both finite classes and infinite classes, or even having at least two classes of different m-degree, can be the word problem of a group. Therefore the question naturally arises as to which ceers can be identified as word problems not only of groups, but of other familiar computably enumerable structures, modulo the several ways of "identifying" equivalence relations, based on natural measures of their relative complexity, we have introduced in Definitions 1.1.1, 1.3.1 and 1.3.4. The contents of this section are meant to be a contribution to this line of research.

We first need of course to specify what we mean by "computably enumerable structures" and their "word problems".

3.1 C.e. algebras

Following the tradition of Mal'cev and Rabin, it is common to postulate that the complexity of the problem of presenting the particular copy of a structure is captured by its atomic diagram. Yet, in algebra one naturally deals with structures whose algebraic structure is easy to describe but it is hard to know whether two terms represent the same element. The paradigmatic example of this phenomenon is the construction, independently due to Boone [20] and Novikov [78], of a finitely presented group with an undecidable word problem. Moreover, the first homomorphism theorem ensures that every countable algebra arises as the quotient of the term algebra on countably many generators. So a countable algebra can *always* be represented in a way in which the complexity of the structure is entirely encoded in its word problem. This motivates the idea, often recurring in the literature, of looking at c.e. structures as given by quotienting $\mathbb{N}$ modulo a ceer. In this paper, we will only be concerned with structures that are algebras.

We recall that a *type of algebras* is a set τ of function symbols, such that each member $f \in \tau$ is assigned a natural number n, called the *arity* of f. An *algebra of type* τ is a pair $\mathcal{A} = \langle A, F \rangle$, where A is a nonempty set, and F is a set of operations on A *interpreting the type*, i.e. in one-to-one correspondence with the function symbols in τ, so that n-ary function symbols of τ correspond to n-ary operations in F.

Definition 3.1.1. An algebra $\mathcal{A}$ of computable type τ, is *computably enumerable* (or, simply, *c.e.*) if there is a triple $\mathcal{A}^- = \langle \mathbb{N}, F, E \rangle$ (called a *positive presentation of* $\mathcal{A}$) such that: (1) F consists of computable operations on $\mathbb{N}$ interpreting the type τ; (2) E is a ceer, which is also a congruence with respect to the operations in F; (3) the quotient $\mathcal{A}_E = \langle \mathbb{N}_E, F_E \rangle$ (called a *positive copy of* $\mathcal{A}$) is isomorphic with $\mathcal{A}$, where $F_E = \{f_E : f \in F\}$, with $f_E([x]_E) = [f(x)]_E$.

For a thorough and clear introduction to c.e. structures see Selivanov's paper [84], where they are called *positive structures*, and Koussainov's tutorial [54].

We will consider c.e. algebras $\mathcal{A} = \langle A, F \rangle$ given by some positive presentation $\mathcal{A}^- = \langle \mathbb{N}, F^-, E \rangle$, and we will work directly with the positive presentation rather than the algebra itself. Thus, if we say $a \in A$ we in fact mean any $a^- \in \mathbb{N}$ such that $[a^-]_E = a$. The ceer E will be often denoted also by $=_{\mathcal{A}}$, as it yields equality in the quotient algebra.

Definition 3.1.2. The *word problem* of a c.e. algebra $\mathcal{A}$ is the ceer $=_{\mathcal{A}}$. (Up to isomorphism of ceers, as in Definition 1.3.1, the word problem is independent of the choice of the positive presentation).

Given a ceer E (having possibly some interesting computational property) it is natural to ask which algebras $\mathcal{A}$ can be positively presented having E as their equality relation $=_{\mathcal{A}}$ (see, e.g., [46, 41]). Surprisingly, much less is known about the reverse problem, namely, given a class of structures $\mathfrak{C}$, which ceers are "realized" by members of $\mathfrak{C}$? This is the main topic of our paper. But, of course, we still need to give a rigorous definition of what we mean by a structure "realizing" a ceer.

We first briefly recap the various ways introduced to indentify ceers (in Definitions 1.1.1, 1.3.1 and 1.3.4). Given ceers R and S, we say that:

1. R is *bi-reducible* to S (denoted by $R \equiv S$) if $R \leq S$ and $S \leq R$;
2. R is *isomorphic* to S (denoted by $R \simeq S$) if $R \leq S$ via a reduction f such that $\mathrm{range}(f) \cap [x]_S \neq \emptyset$, for all x;
3. R is *strongly isomorphic* to S (denoted $R \simeq_{\mathrm{s}} S$) if $R \leq S$ via a computable permuation of $\mathbb{N}$.

Definition 3.1.3. (i) If R is a ceer, $\mathcal{A}$ is a c.e. algebra, and $\approx \in \{\equiv, \simeq, \simeq_{\mathrm{s}}\}$ then we say that R is $\approx$*-realized by* $\mathcal{A}$ if $R \approx =_{\mathcal{A}}$. (Recall that $=_{\mathcal{A}}$ denotes the word problem of $\mathcal{A}$.)

(ii) A class $\mathfrak{C}$ of algebras of the same type is *$\approx$-complete for a class $\mathbb{C}$ of ceers* (where $\approx \in \{\equiv, \simeq, \simeq_{\mathrm{s}}\}$) if every ceer in $\mathbb{C}$ is $\approx$-realized by some c.e. copy of an algebra from $\mathfrak{C}$. We simply say that $\mathfrak{C}$ is *$\approx$-complete for the ceers* if $\mathfrak{C}$ is $\approx$-complete for the class of all ceers.

It is trivial to observe the following facts.

Fact 3.1.4. *Let $\mathbb{C}$ be a class of ceers and $\mathfrak{C}$ be a class of algebras.*

(i) If $\mathfrak{C}$ is $\simeq$-complete for $\mathbb{C}$, then it is $\equiv$-complete for $\mathbb{C}$; if $\mathfrak{C}$ is $\simeq_s$-complete for $\mathbb{C}$, then it is $\simeq$-complete for $\mathbb{C}$-

(ii) If all members of $\mathbb{C}$ have no finite equivalence classes, then $\mathfrak{C}$ is $\simeq$ complete for $\mathbb{C}$ if and only if $\mathfrak{C}$ is $\simeq_s$-complete for $\mathbb{C}$.

Proof. Item (i) follows immediately from the definitions of bi-reducibility, isomorphism and strong isomorphism among ceers, while item (ii) is an immediate corollary of Fact 1.3.5. □

3.2 Classes of algebras that are complete for the ceers

We now begin to look at some natural classes of c.e. algebras in relation to the problem of $\approx$-completeness for ceers, with $\approx \in \{\equiv, \simeq, \simeq_{\mathrm{s}}\}$. Our examples of c.e. algebras will be more conveniently introduced via the notion of a computably enumerable presentation. In a variety of algebras with finite or countable type, if the *term algebra* $T(X)$ on a finite or countable set X (see e.g. [24, §10]) exists (existence is guaranteed if, as in our future examples, X is nonempty) then, up to isomorphisms, $T(X)$ can be presented as a computable algebra: we may assume that X is decidable, $T(X)$ has decidable universe (which is infinite in all our examples), computable operations, and equality is syntactic equality. If, in addition the identities of the variety form a c.e. binary relation on $T(X)$, then we have the following definition.

Definition 3.2.1. In a variety as above, a *c.e. presentation* is a pair $\mathcal{A} = \langle X; R \rangle$ where X is a set, R is a binary relation on $T(X)$, and $\mathcal{A}$ denotes the quotient algebra $T(X)_{/N_R}$, where N_R is the c.e. congruence on $T(X)$ generated by R together with the identities of the variety. An algebra $\mathcal{A}$

of the variety is *c.e. presented* (*c.e.p.*), if it is of the form $\langle X; R\rangle$ as just described.

A special case is provided by *finite presentations*, where both X and R are finite.

The following fact is well known:

Lemma 3.2.2. *In a variety as above, an algebra is c.e. if and only if it is isomorphic to some c.e.p. algebra.*

Proof. We sketch the proof. If $\mathcal{A} = \langle X; R\rangle$ is a c.e. presentation, then there is a computable isomorphism f of $T(X)$ with an algebra having $\mathbb{N}$ as universe, and equipped with a set F of suitable computable functions corresponding, via the isomorphism, to the operations of $\mathcal{A}$. Then $\mathcal{B} = \langle \mathbb{N}, F, E\rangle$ is a positive presentation of $\mathcal{A}$, where E is the ceer corresponding under the isomorphism to the c.e. relation N_R on $T(X)$. Notice that according to Definition 3.1.2, equality $=_{\mathcal{B}}$ of $\mathcal{B}$ coincides with E.

For the converse, assume that $\mathcal{A} = \langle \mathbb{N}, F, E\rangle$ is a positive presentation. By the universal property of $T(\mathbb{N})$ (namely, the term algebra on the set $\mathbb{N}$ of generators), there is a unique epimorphism $\nu : T(\mathbb{N}) \to \mathcal{A}_E$ which commutes with the mapping $x \mapsto [x]_E$ from $\mathbb{N}$ to $\mathcal{A}_E$, and the insertion of generators $x \mapsto x$ from $\mathbb{N}$ to $T(\mathbb{N})$. Namely, if $p(x_1, \dots, x_k) \in T(\mathbb{N})$ is a term, and p_F interprets p using the operations in F, then $\nu(p(x_1, \dots, x_k)) = [p_F(x_1, \dots, x_k)]_E$, by the properties of E. It follows that the kernel R of ν is a c.e. binary relation on $T(X)$, and by universal algebra, the c.e. presentation $\langle \mathbb{N}; R\rangle$ is isomorphic with $\mathcal{A}_E$. $\square$

To describe some of the consequences of Lemma 3.2.2 which are relevant to our later examples, we first generalize Definition 1.3.1 to *partial ceers*, i.e. c.e. equivalence relations having as domains c.e. subsets of $\mathbb{N}$. If R, S are partial ceers with domains X, Y respectively, we say that R and S are isomorphic ($R \simeq S$: we use the same symbol as in Definition 1.3.1) if there is partial computable function $g : X \to Y$ whose domain contains X, such that $x\ R\ y$ if and only if $g(x)\ S\ g(y)$ for all $x, y \in X$, and range(g) intersects all S-equivalence classes.

One direction of the proof of the previous lemma actually shows that every c.e. presentation $\langle X; R\rangle$ has a positive presentation $\langle \mathbb{N}, F, E\rangle$ such that $N_R \simeq E$ as partial ceers, as witnessed by the computable isomorphism $f : T(X) \to \mathbb{N}$. The other direction of the proof shows in fact that for every positive presentation $\mathcal{A} = \langle \mathbb{N}, F, E \rangle$ there is a c.e. presentation $\langle \mathbb{N}; R \rangle$ which is isomorphic to $\mathcal{A}_{/E}$, and $R \simeq E$ as partial ceers. This follows from the fact that ν is onto, and therefore the computable mapping $p(x_1, \ldots, x_k) \mapsto p_F(x_1, \ldots, x_k)$ provides a reduction from R to E whose range intersects all E-equivalence classes.

3.2.1 The word problem as a ceer on terms, or as a ceer on the free algebra

When trying to show that some ceer S is $\simeq$-realized by a c.e. presentation $\langle X; R \rangle$, the above remarks suggest, in accordance to many algebra textbooks (see e.g. [24, p.252]) to consider N_R as the word problem of the c.e. presentation, and show that $S \simeq N_R$ as partial ceers. This is fully consistent with Definition 3.1.2, since, as we have seen, $S \simeq E$, where E is the ceer of the positive presentation assigned to $\langle X; R \rangle$ in the proof of Lemma 3.2.2.

In fact, our examples of c.e. algebras will come from varieties (such as semigroups, monoids, groups, rings) in which the identities of the variety generate a decidable congruence I on $T(X)$. By decidability of I, we mean that we can fix a computable mapping $p \mapsto \overline{p} : T(X) \to T(X)$ with decidable range, picking up exactly one element in each I-equivalence class, so that the free algebra $F(X)$, taken to be $T(X)_{/I}$, can be presented as a computable algebra having this range as universe. Let now $\langle X; R \rangle$ be a c.e. presentation. By universal algebra, there is a c.e. congruence $\overline{R}$ on $F(X)$ (namely, $\overline{R} = N_{R/I}$, using common notation in universal algebra) such that $T(X)_{/N_R}$ is isomorphic with $F(X)_{/\overline{R}}$ and

$$p \; N_R \; q \Leftrightarrow \overline{p} \; \overline{R} \; \overline{q},$$

for every $p, q \in T(X)$. This gives an isomorphism of partial ceers between N_R and $\overline{R}$. Conversely, given a binary c.e. relation $\overline{R}$ on $\overline{X} = \{\overline{x} : x \in X\}$, then one can find a c.e. congruence R on $T(X)$ such that $T(X)_{/R}$ is isomorphic with $F(X)_{/N_{\overline{R}}}$, where $N_{\overline{R}}$ is the c.e. congruence on $F(X)$ generated by $\overline{R}$. Moreover, R and $N_{\overline{R}}$ are isomorphic as partial ceers.

This suggests to adopt, in these varieties, even a more simplified, yet equivalent, approach to word problems of c.e. algebras, and agree that a c.e. presentation is a pair $\langle X; R \rangle$ where R is a binary c.e. relation on $F(X)$ and, in this case, $\langle X; R \rangle$ denotes the quotient $F(X)_{/N_R}$, where N_R is the congruence generated on $F(X)$ by R, and we take N_R as the word problem of the c.e. algebra so presented. Of course, in general the elements of $F(X)$ will not be presented directly as certain elements of $T(X)$ but in some simplified "normal form", obtaining in any case a computably isomorphic copy of the free algebra, and up to isomorphism of partial ceers, the same word problem.

3.2.2 Semigroups

Throughout the section our references for terminology about semigroups and monoids are the textbooks [29] and [49]. In view of Definition 3.2.1 (and the subsequent adjustment in Subsection 3.2.1), towards an explicit description of a c.e.p. semigroup it is sufficient to describe what the free semigroup $F(X)$ on X and the c.e. binary relation R on $F(X)$ are. Hence, we recall that the free semigroup on a set X can be taken to be $\langle X^* \smallsetminus \{\lambda\}, \cdot\rangle$, where in general Y^* denotes the collection of finite words of letters from a set Y, λ is the empty string, and $\cdot$ is concatenation of words.

Definition 3.2.3. A semigroup S is a *right-zero band* if $ab = b$ for all $a, b \in S$.

Theorem 3.2.4. *The class of right-zero bands is $\simeq$-complete for the ceers.*

Proof. Let R be a given ceer, and fix a computable set $X = \{x_i : i \in \mathbb{N}\}$ of generators. Consider the c.e. binary relation $\widehat{R}$ on $F(X)$:

$$\widehat{R} = \{x_i = x_j : i \; R \; j\} \cup \{x_j x_i = x_i : i, j \in \mathbb{N}\}.$$

Let $S = \langle X; \widehat{R}\rangle$ be the c.e.p. semigroup so presented. In particular notice that $ux_i =_S x_i$ for any word u and any generator x_i.

It is easy to see that

$$i \; R \; j \Leftrightarrow x_i =_S x_j,$$

so that $R \leq =_S$ by the reduction $f(i) = x_i$. On the other hand, as the range of f intersects all $=_S$-equivalence classes (since $u =_S x_i$ where x_i is the last bit of u, as follows from the relations), we have that $=_S \simeq R$. □

Of course the same result holds if we replace right-zero bands with *left-zero bands*, i.e. semigroups in which $ab = a$ for all pairs a, b.

3.2.3 Monoids

Next, we consider the case of monoids. Recall in this case that the free monoid $F(X)$ on X can be taken to be $\langle X^*, \cdot \rangle$, where again $\cdot$ is concatenation.

Definition 3.2.5. A monoid $M = \langle M, \cdot \rangle$ is *right-zero band-like* if $ab = b$ for every $a, b \in M \smallsetminus \{1\}$ (where 1 denotes the identity element).

Theorem 3.2.6. *The class of right-zero band-like monoids is $\simeq$-complete for the ceers.*

Proof. Let R be a given ceer, and fix again a computable set $X = \{x_i : i \in \mathbb{N}\}$ of generators. Consider the c.e. binary relation $\widehat{R}$ on $F(X)$:

$$\widehat{R} = \{x_i = x_j : i \; R \; j\} \cup \{x_j x_i = x_i : i \in \mathbb{N} \smallsetminus \{0\},\ j \in \mathbb{N}\} \cup \{x_0 = \lambda\}.$$

Let $M = \langle X; \widehat{R} \rangle$ be the c.e.p. monoid so presented. The proof that $R \simeq =_M$ is as in the proof of Theorem 3.2.4. $\square$

Again, right-zero band-like monoids can be replaced by left-zero band-like monoids in the result above.

3.3 Classes of algebras that are not complete for the ceers

We try in this section to identify algebraic properties that prevent classes of algebras sharing these properties to be $\approx$-complete for the ceers, with $\approx \in \{\equiv, \simeq, \simeq_s\}$.

3.3.1 Semigroups

For our first observation, we need the following definition.

Definition 3.3.1. A semigroup S is *periodic* if, for all $a \in S$, there are numbers $1 \leq n < m$ such that $a^n = a^m$.

Recall that a ceer R is *dark* if R has infinitely many equivalence classes but it does not admit any infinite c.e. transversal, i.e. an infinite c.e. set W such that if $x, y \in W$ and $x \neq y$ then $x \not R y$. We have discussed the existence and properties of dark ceers in Section 1.2.

Theorem 3.3.2. *The class of semigroups which are not periodic is not $\equiv$-complete for the ceers.*

Proof. Let S be a non-periodic c.e. semigroup. Then there exists an element $a \in S$ such that $a^m \neq_S a^n$ if $m \neq n$. Thus $\{a^n : n \in \mathbb{N}\}$ is an infinite c.e transversal, implying that $=_S$ cannot $\equiv$-realize any dark ceer, as the property of having an infinite c.e. transversal is invariant under bi-reducibility. $\square$

Recall that a *diagonal function* for an equivalence relation R is a computable function d such that $d(x) \not R x$, for every x. The next theorem identifies a natural class of semigroups which are not $\simeq$-complete for the ceers. Examples of semigroups filling the description in the statement of the theorem are for instance the semigroups without idempotent elements.

Theorem 3.3.3. *The class of semigroups S for which there exists a number n such that $x^n \neq_S x$ for every x is not $\simeq$-complete for the ceers.*

Proof. Suppose S is a c.e. semigroup as in the statement of the theorem. Take any x, and define $d(x) = x^n$. Then d is a diagonal function for $=_S$, and thus S cannot $\simeq$-realize any ceer which does not possess a diagonal function, such as for instance the precomplete ones, see Definition 1.4.2 and Corollary 1.4.4 (or the survey paper [2]). $\square$

3.3.2 Monoids

We now take a quick look at monoids.

Definition 3.3.4. Let M be a monoid. A non-unit element $x \in M$ is a *torsion element* if there exists a number $n > 0$ such that $x^n = 1$; otherwise x is *non-torsion*. Moreover, a monoid is said to be *torsion* if every element is a torsion element, *non-torsion* otherwise.

We observe:

Theorem 3.3.5. *The class of non-torsion monoids is not $\equiv$-complete for the ceers.*

Proof. Let $x \in M$ be a non-torsion element. Then $\{x, x^2, (x^2)^2, \ldots\}$ is an infinite c.e. transversal for $=_M$. The proof is now similar to the proof of Theorem 3.3.2. $\square$

3.4 On finitely presented semigroups and a question of Gao and Gerdes

After having looked at algebraic properties preventing completeness, we now move to investigate what can happen if we restrict the complexity of the presentation of the structure. In particular, in the following we will focus mainly on finitely generated and finitely presented semigroups.

We recall the following observation of Gao and Gerdes [45, p.58] (where the statement refers to finitely presented groups, but it is obviously extendable to all groups).

Fact 3.4.1. *The class of groups is not $\equiv$-complete for the ceers.*

Proof. If R is an undecidable ceer with only finitely many undecidable equivalence classes (it is easy to see that there are even undecidable ceers with only finite equivalence classes: for instance, there are dark ceers with only finite classes, see Proposition 1.2.5 (or [5, Corollary 4.15]) then there cannot be any c.e. group G such that $=_G \equiv R$: for otherwise, by the reduction $=_G \leq R$ we would have that either $=_G$ is finite, and thus $R \not\leq =_G$, or there are decidable $=_G$-classes, but as observed in the introduction all $=_G$-equivalence classes are computably isomorphic with each other, which would imply that $[1]_G$ is decidable and thus $=_G$ is decidable($u =_G v$ if and only if $uv^{-1} \in [1]_{=_G}$), giving that R is decidable by the reduction $R \leq =_G$. $\square$

For this reason, Gao and Gerdes (see [45, Problem 10.3]) ask whether the class of f.p. semigroups is $\equiv$-complete for the ceers. This is an interesting question, motivated by a celebrated theorem due to Shepherdson [86] stating that if $\{A_i : i \in \mathbb{N}\}$ is a uniformly c.e. sequence of c.e. sets (meaning that the relation "$x \in A_i$", in i, x, is c.e.), B is a c.e. set, and the relation "$x \in A_i$" is $\leq_T B$, then there is a f.p. semigroup S with the following three properties: (1) there is an effective correspondence $w_i \mapsto A_i$ between a c.e. set $\{w_i : i \in \mathbb{N}\}$ of words and $\{A_i : i \in \mathbb{N}\}$ so that, effectively in i, one can find Turing reductions establishing $[w_i]_{=_S} \equiv_T A_i$; (2) the Turing degrees of the various classes $[w]_{=_S}$ consist of the least Turing degree, together with all finite joins of the various degrees $\deg_T(A_i)$; (3) $=_S \equiv_T B$.

The next theorem will provide a negative answer to Gao and Gerdes' question.

Theorem 3.4.2. *Suppose that $\{E_i : i \in \mathbb{N}\}$ is a uniformly c.e. sequence of ceers such that the set $\{i : E_i$ is finite$\}$ is c.e. Then there exists an infinite ceer E such that for every i,*

$$E_i \not\leq E \text{ or } E_i \text{ is finite (i.e. } E_i \text{ has finitely many classes).}$$

In particular, for every i, $E_i \not\equiv E$.

Proof. Let $V = \{i : E_i$ is finite$\}$ be c.e., and let $\{V_s : s \in \mathbb{N}\}$ be a c.e. approximation to V, that is, a strong array of finite sets, with $V_s \subseteq V_{s+1}$ for every s, and $V = \bigcup_s V_s$. Let also $\{\phi_j : j \in \mathbb{N}\}$ be an acceptable indexing of the partial computable functions.

Our desired ceer must satisfy the following requirements:

$$\begin{aligned} \mathcal{P}n : &\qquad E \text{ has at least } n+1 \text{ classes,} \\ \mathcal{Q}\langle i,j\rangle : &\qquad E_i \not\leq E \text{ via } \varphi_j, \text{ or } E_i \text{ is finite.} \end{aligned}$$

We order the requirements according to the priority ordering:

$$\mathcal{P}_0 < \mathcal{Q}_0 < \ldots < \mathcal{P}n < \mathcal{Q}_n < \ldots <$$

We say that $\mathcal{R}$ has *higher priority than* $\mathcal{R}'$ (or $\mathcal{R}'$ has *lower priority than* $\mathcal{R}$) if $\mathcal{R} < \mathcal{R}'$.

We construct E in stages. At stage s we define an equivalence relation E_s, so that: $E_0 = \mathrm{Id}$ (the identity ceer); for every s, $E_s \subseteq E_{s+1}$, E_s is a finite extension of Id (the identity ceer) and, uniformly in s, $E_s \smallsetminus \mathrm{Id}$ can be presented by its canonical index; and finally $E = \bigcup_s E_s$ is our desired equivalence relation. E_{s+1} will be generated by E_s plus finitely many pairs of numbers which are, as we say, *E-collapsed* at $s+1$.

The strategy to satisfy $\mathcal{P}_n$ consists in picking $n+1$ numbers which are still pairwise E-non-equivalent, and restraining their equivalence classes from future E-collapses.

The strategy to satisfy $\mathcal{Q}_{\langle i,j\rangle}$ goes as follows. At a given stage s, we say that *evidence appears that ϕ_j is not a reduction from E_i to E* if one of the following happens:

(A) ϕ_j does not look total, i.e. we see some witness v such that ϕ_j is still undefined on v;

(B) we see two witnesses x, y such that $\phi_j(x)$ and $\phi_j(y)$ both converge, and at the given stage $x \not{E}_i y$, but already $\phi_j(x)\ E\ \phi_j(y)$.

Notice that, contrary to what one may expect, we do not bother to seek evidence given by two witnesses x, y such that $\phi_j(x)$ and $\phi_j(y)$ both converge, and at the given stage already $x \; E_i \; y$, but $\phi_j(x) \not{E} \phi_j(y)$. Indeed, our action on trying to meet $\mathcal{Q}_{\langle i,j \rangle}$ will force the opponent to give up on totality of ϕ_i, or leave non-E_i-equivalent two numbers whose ϕ_j-images we have already E-collapsed.

Notice that, independently of our will, evidence due to (A) may be lost at a later stage t, if $\phi_{j,t}(v) \downarrow$; evidence due to (B) may be lost at a later stage t if $x \; E_{i,t} \; y$.

Here is the description of our strategy in isolation:

(1) we wait to see $i \in V$; if i gets enumerated into V then the requirement is satisfied, so we stop worrying about it, and definitively move on to satisfy the lower priority requirements;

(2) while waiting to see $i \in V$ or for evidence to appear that ϕ_j is not a reduction as in (B), we threaten to make E finite by E-collapsing all the elements $\geq m$, where m is a threshold indicated to $\mathcal{Q}_{\langle i,j \rangle}$ by the restraint placed by higher priority requirements;

(3) while waiting to see $i \in V$, if evidence has appeared that ϕ_j is not a reduction as in (B) then

   (a) while this evidence persists, we move on to satisfy the lower priority requirements;

   (b) when this evidence gets lost, we loop back to (2).

The outcomes of the strategy are evident: moving out of (1) is a finitary outcome satisfying the requirement, as E_i is finite.

If (1) does not show up, then we claim that we cannot loop between (3b) and (2) infinitely often. For otherwise ϕ_j would be total, E finite (as we E-collapse all $x, y > m$), but then ϕ_j cannot be an injective reduction from the equivalence classes of E_i (which is infinite) to the equivalence classes of E (which would be finite). Therefore our strategy eventually stops at (3b) because of (A) (outcome: ϕ_j is not total), or because of (B) (outcome: $x \not{E}_i y$ but $\phi_j(x) \; E \; \phi_j(y)$ for some x, y).

Since all strategies have finite outcomes, the conflicts between different strategies are resolved by a straightforward finite priority argument.

The construction At each stage, requirements may be initialized, and they are so at stage 0; or, in case of $\mathcal{Q}$-requirements, they may be declared permanently satisfied in which case they are met once and for all.

The construction makes use at each stage of the following parameters for every requirement $\mathcal{R}$: if $\mathcal{R}$ is initialized, then these parameters are undefined. The parameter $m^{\mathcal{R}}(s)$ denotes the restraint imposed at stage s by $\mathcal{R}$, with $\mathcal{R} \in \{\mathcal{P}\mathcal{Q}\}$, to lower priority requirements, so that they can only E-collapse pairs of elements $x, y > m^{\mathcal{R}}(s)$.

The parameter $M(\langle i, j\rangle, s)$ (if $\mathcal{Q}_{\langle i,j\rangle}$ is not initialized, and thus we may suppose $s > 0$), is defined as follows: if there is $v \leq s$ such that either

1. $\phi_{j,s}(v)\uparrow$, or
2. $v = \langle x, y\rangle$ and $\phi_{j,s}(x)$ and $\phi_{j,s}(y)$ both converge and $x \not{E}_{i,s} y$, but $\phi_{j,s}(x)\ E_{s-1}\ \phi_{j,s}(y)$.

then let $M(\langle i, j\rangle, s) = \langle v, 0\rangle$ in the former case, otherwise $M(\langle i, j\rangle, s) = \langle v, 1\rangle$. Let $M(\langle i, j\rangle, s) = \langle 0, 2\rangle$ if there exists no such v.

If not otherwise specified, at each stage $s > 0$ each parameter maintains the same value as at the previous stage, or stays undefined if it was undefined at the previous stage.

We say that $\mathcal{P}_n$ *requires attention at* $s + 1$ if it is initialized.

We say that $\mathcal{R} = \mathcal{Q}_{\langle i,j\rangle}$ *requires attention at* $s + 1$ if $\mathcal{Q}_{\langle i,j\rangle}$ has not as yet been declared permanently satisfied and (in order):

(1) $\mathcal{Q}_{\langle i,j\rangle}$ is initialized; or

(2) $i \in V_s$; or

(3) $M(\langle i, j\rangle, s + 1) \neq M(\langle i, j\rangle, s)$.

Stage 0 Initialize all requirements, and set $m^{\mathcal{R}}(k, 0)$ and $M(k, 0)$ undefined for all $\mathcal{R} \in \{\mathcal{P}, \mathcal{Q}\}$, and $k \in \mathbb{N}$. Let $E_0 = \mathrm{Id}$.

Stage $s + 1$ Let $\mathcal{R}$ be the least requirement that requires attention: there is such a least requirement since almost all requirements are initialized when we begin stage $s + 1$.

Case 1 If $\mathcal{R} = \mathcal{P}_n$ then $\mathcal{R}$ is initialized: pick the least $n+1$ numbers bigger than any number so far used in the construction (thus these numbers are still non-E-equivalent) and let $m^{\mathcal{R}}(s+1)$ be the greatest one of the numbers which have been picked; $\mathcal{R}$ stops being initialized.

Case 2 Suppose that $\mathcal{R} = \mathcal{Q}_{\langle i,j \rangle}$. We refer to the various cases for which $\mathcal{R}$ may require attention:

(a) (Case (1) of requiring attention) let $m^{\mathcal{R}}(s+1) = \max\{m^{\mathcal{R}'}(s) : \mathcal{R}' < \mathcal{R}\}$ (notice that no $\mathcal{R}' < \mathcal{R}$ is initialized), so that $\mathcal{R}$ stops being initialized;

(b) (Case (2) of requiring attention) declare $\mathcal{R}$ *permanently satisfied* (and will stay so forever);

(c) (Case (3) of requiring attention) E-collapse all x, y such that $m^{\mathcal{R}}(s+1) < x, y \leq s$;

Whatever the case, initialize all $\mathcal{R}' > \mathcal{R}$, by setting $m^{\mathcal{R}'}(s+1)\uparrow$, and $M(\langle i', j' \rangle, s)\uparrow$ if $\mathcal{Q}\langle i', j' \rangle > \mathcal{R}$.

Let E_{s+1} be the equivalence relation generated by E_s plus the pairs of numbers which have been E-collapsed at $s+1$.

Verification The verification is based on the following lemma.

Lemma 3.4.3. *For every requirement $\mathcal{R}$, $\mathcal{R}$ is initialized only finitely many times, $m^{\mathcal{R}} = \lim_s m^{\mathcal{R}}(s)$ exists, $\mathcal{R}$ eventually stops requiring attention, and $\mathcal{R}$ is met.*

Proof. Suppose that the claim is true of every $\mathcal{R}' < \mathcal{R}$, and let s_0 be the greatest stage at which some $\mathcal{R}' < \mathcal{R}$ has received attention, with $s_0 = 0$ if $\mathcal{R} = \mathcal{P}0$. Let $m = \max\{m^{\mathcal{R}'} : \mathcal{R}' < \mathcal{R}\}$.

At the beginning of stage s_0+1, $\mathcal{R}$ is initialized, and thus requires attention, acts through (1) or (2a), and after this stage it will never be initialized again.

Case $\mathcal{R} = \mathcal{P}n$. If $\mathcal{R} = \mathcal{P}n$, then $\mathcal{R}$ acts, picks $n+1$ unused numbers. These numbers are still E-non-equivalent. $\mathcal{R}$ defines a value of $m^{\mathcal{R}}(s_0+1)$ which will never change hereafter, and thus is the limit value $m^{\mathcal{R}}$ of $m^{\mathcal{R}}(s)$. This limit value sets a restraint on lower priority requirements which therefore can never E-collapse any pair of these $n+1$ numbers. This shows also that $\mathcal{P}n$ is met, as the final E has at least $n+1$ equivalence classes.

Case $\mathcal{R} = \mathcal{Q}\langle i,j\rangle$. At stage s_0+1, $\mathcal{Q}\langle i,j\rangle$ defines the last value $m^{\mathcal{R}} = m^{\mathcal{R}}(s_0+1)$ of its parameter $m^{\mathcal{R}}$: notice that this value will never change again, and is in fact the same as $m^{\mathcal{R}'}$, where $\mathcal{R}'$ is the P-requirement immediately preceding $\mathcal{R}$ in the priority ordering. If $\mathcal{R}$ receives attention at some stage $s_1+1 > s_0+1$ and acts through Case (2b), then the action declares $\mathcal{R}$ permanently satisfied, $\mathcal{R}$ will never receive attention again, E_i is finite then $\mathcal{R}$ is met.

If we exclude action Case (2b) after s_0+1, then E_i is infinite. We claim that still $\mathcal{R}$ requires attention finitely many times after s_0+1. For otherwise, at infinitely many stages s we E-collapse all numbers $m^{\mathcal{R}} < x, s \leq s$, and therefore E is finite since we E-collapse all $x, y > m^{\mathcal{R}}$. On the other hand E_i is infinite, so ϕ_j cannot induce a 1-1 mapping from E_i-equivalence classes to E-equivalence classes, thus eventually $M(\langle i,j\rangle, s)$ stabilizes on a value $\langle v,k\rangle$ with $k \in \{0,1\}$ and stops receiving attention again: contradiction. So (if we never act through Case (2b)) we are forced to conclude that $M(\langle i,j\rangle, s)$ stabilizes on some $\langle v,0\rangle$, and thus ϕ_j is not total, $\mathcal{R}$ is satisfied, and $\lim_s m^{\mathcal{R}}(s) = m$; or it stabilizes on some $\langle\langle x,y\rangle,1\rangle$, in which case $x \not{E}_i y$ and $\phi_j(x)\ E\ \phi_j(y)$, and $\mathcal{R}$ is met. $\square$

$\square$

Corollary 3.4.4. *No class $\mathfrak{A}$ of finitely generated semigroups is $\equiv$-complete for the ceers.*

Proof. Up to computable isomorphisms, we can assume that a finitely generated c.e.p. semigroup is of the form $\langle\{0,1,\ldots,n\}, R\rangle$ where R is a c.e. subset of $(\{0,1,\ldots,n\}^*)^2$. Let f be a computable function such that $\{V_{f(n,i)} : i \in \mathbb{N}\}$ computably lists all c.e. subsets of $(\{0,1,\ldots,n\}^*)^2$. From this we get a computable listing $\{S_{\langle n,i\rangle} : n,i \in \mathbb{N}\}$ (where $S_{\langle n,i\rangle} = \langle\{0,1,\ldots,n\}^*, V_{f(n,i)}\rangle$) of all finitely generated c.e.p. semigroups, and a corresponding computable listing $\{E_{\langle n,i\rangle} : n,i \in \mathbb{N}\}$ of their word problems.

In view of the previous theorem it suffices to show that $\{\langle n, i\rangle : E_{\langle n,i\rangle} \text{ finite}\}$ is c.e. Let $X = \{0, 1, \ldots, n\}$. We claim that $E_{\langle n,i\rangle}$ is finite if and only if

$$(\exists m > 0)(\forall \sigma \in X^*)[|\sigma| = m \Rightarrow (\exists \tau \in X^*)[|\tau| < |\sigma| \,\&\, \tau \; E_{\langle n,i\rangle} \; \sigma]], \qquad (\star)$$

which is a c.e. expression (in which for a given string ρ, the symbol $|\rho|$ denotes the length of ρ). On the one hand, if $E_{\langle n,i\rangle}$ is finite, one can fix a finite transversal A which meets all the equivalence classes of $E_{\langle n,i\rangle}$. Since each word of $S_{\langle n,i\rangle}$ is equivalent to a word from A, we have that $(\star)$ holds for m, where $m = \max\{|\sigma| : \sigma \in A\} + 1$.

On the other hand, assume that $(\star)$ holds, and fix such an m. We claim in this case that every word is $E_{\langle n,i\rangle}$-equivalent to some word of length $\leq m$. Towards a contradiction, let $n > m$ be the least number such that there exists σ with $|\sigma| = n$, and $[\sigma]_{E_{\langle n,i\rangle}}$ contains no word of length $\leq m$. Now, let $\sigma_0 = \sigma \restriction m$ (i.e., the initial segment of σ of length m), and let σ_1 be such that $\sigma = \sigma_0\sigma_1$. Then σ_0 is $E_{\langle n,i\rangle}$-equivalent to some ρ with $|\rho| < m$. Therefore, by definition of $\langle X; E_{\langle n,i\rangle}\rangle$, we have that $\sigma = \sigma_0\sigma_1 E_{\langle n,i\rangle} \rho\,\sigma_1$, but $|\rho\,\sigma_1| < n$, contradicting the minimality of n. □

As a particular case, this provides a negative solution to Gao and Gerdes' question:

Corollary 3.4.5. *The class of f.p. semigroups is not $\equiv$-complete for the ceers.*

Proof. Immediate. □

Next, we observe that, given a f.p. semigroup S, the number of finite and infinite equivalence classes of the word problem $=_S$ gives us some information about the ceers realized by S. We basically owe the following arguments to [19], see also [59].

Lemma 3.4.6. *If S is a f.p. semigroup then there is a partial computable function ψ such that for every word w, $\psi(w)\downarrow$ if and only if the $=_S$-equivalence class of w is finite, and, when convergent, $\psi(w)$ outputs the canonical index of the equivalence class $[w]_{=_S}$ of w.*

Proof. Given a word w, we can effectively generate its $=_S$-equivalence class in a treelike fashion as follows. The root of the tree is w. Each node u has as children the words that can be obtained from u using the relations and which have not yet appeared as a node in the path from the root to the present node. Note that one relation produces only finitely many children, and there are only finitely many relations: hence, this is a finitely branching tree. By the König Lemma if the equivalence class of w is finite, we eventually stop generating new nodes on any branch of the tree: when this happen we have generated the entire equivalence class of w, and we can compute the canonical index of this class. $\square$

Theorem 3.4.7. *Let S be a f.p. semigroup.*

(i) If S has finitely many infinite equivalence classes, then $=_S$ is decidable. Therefore no undecidable ceer can be $\equiv$-realized by such an S.

(ii) If S has infinitely many finite equivalence classes, then $=_S$ is light. Therefore, neither finite nor dark ceers can be $\equiv$-realized by such an S.

Proof. Suppose that $=_S$ has only finitely many infinite equivalence classes. Assume that $\{v_i : i \in I\}$ is a finite set of words, with $v_i \neq S v_j$ if $i \neq j$, spanning these infinite equivalence classes. Given words x, y, generate the equivalence classes of x and y in a tree-like fashion as in the proof of Lemma 3.4.6, until one of the following happens: (1) $[x]_{=_S}$ and $[y]_{=_S}$ cannot grow any more (and we can decide this, as explained in the proof Lemma 3.4.6); (2) some v_i is generated in one equivalence class, and some v_j is generated in the other one; (3) some v_i is generated in one of the two equivalence classes and the other one has stopped (again, we can decide this latter outcome). In any case we can decide if the two words are equal. This proves statement (i).

Now, we prove (ii). Let S be a f.p. semigroup with infinitely many finite equivalence classes and let ψ be the partial computable function of Lemma 3.4.6. Using ψ we can build in stages an infinite c.e. transversal $\{a_0, a_1, \dots\}$ for $=_S$:

Step 0. Let w_0 be the first word such that $\psi(w_0)\downarrow$ and define a_0 to be the least element of the finite set $D_{\psi(w_0)}$.

Step $n+1$. Let w_{n+1} be the first word such that $\psi(w_{n+1})\downarrow$ and

$$D_{\psi(w_{n+1})} \cap (\bigcup_{i \leq n} D_\psi(w_i)) = \emptyset$$

and let a_{n+1} be the least element of $D_{\psi(w_{n+1})}$. $\square$

3.5 Transversals of word problems of finitely generated semigroups

Following the line of the previous section, we single out several classes of ceers which cannot be $\equiv$-realized by any finitely generated semigroup: see Theorem 3.5.12.

3.5.1 More immunity and darkness notions

In Section 1.2, we have seen that dark ceers can be conveniently characterized by the notion of a transversal for an equivalence relation, introduced in Definition 1.2.2: in fact, a ceer R is dark if and only if it admits no infinite c.e. tranversal, or equivalently that every infinite transversal of R is immune (see Definition 1.1.3).

Other stronger immunity notions have been widely considered in classical computability theory, and we briefly review their definitions.

Definition 3.5.1. We say that a set $X \subseteq \mathbb{N}$ *is intersected* by a disjoint sequence (or *array*) $(X_n)_{n\in\mathbb{N}}$ of subsets of $\mathbb{N}$ (*disjoint* means that $X_n \cap X_m = \emptyset$ if $n \neq m$) if, for all n, $X_n \cap X \neq \emptyset$. An infinite set is *hyperimmune* if it is not intersected by any *strong disjoint array*, i.e. a disjoint array $(F_n)_{n\in\mathbb{N}}$ of finite sets, presented by their canonical indices: $F_n = D_{f(n)}$ for some computable function f. A c.e. set is called *hypersimple* if its complement is immune.

Similarly, an infinite set is *hyperhyperimmune* if it is not intersected by any *weak disjoint array*, i.e. a disjoint array $(F_n)_{n\in\mathbb{N}}$ of finite sets, presented by their c.e. indices: $F_n = W_{f(n)}$ for some computable function f. A c.e. set is *hyperhypersimple* if its complement is hyperhyperimmune.

This suggests also the following definition.

Definition 3.5.2. A ceer is *hyperdark* (respectively, *hyperhyperdark*) if all its infinite transversals are hyperimmune (respectively, hyperhyperimmune).

Clearly every hyperhyperdark ceer is also hyperdark, and every hyperdark ceer is also dark.

We have already seen in Section 1.2 that an obvious way to give examples of dark ceers is to consider unidimensional ceers generated by any simple set (unidimensional ceers have been introduced in Definition 1.1.2). We can obtain easy examples of hyperdark and hyperhyperdark ceers similarly, considering any hypersimple or, respectively, hyperhypersimple set, instead: since a transversal T of R_X satisfies that $T \smallsetminus X^c$ has at most one element (where X^c denotes the complement of X), and infinite subsets of immune (respectively, hyperimmune, hyperhyperimmune) sets are immune (respectively: hyperimmune, hyperhyperimmune), it is easy to see that R_X is dark (respectively: hyperdark, hyperhyperdark) if and only if X is simple (respectively: hypersimple, hyperhypersimple). Using the known facts that there exist sets which are hyperhypesimple but not hypersimple, and sets which are hypersimple but not simple, one easily sees that the unidimensional ceers are also enough to witness that the inclusions between the classes of dark, hyperdark and hyperhyperdark ceers are proper. More interesting examples of ceers lying in these classes will be given in Section 3.5.3.

3.5.2 Π^0_1 classes and $\equiv$-realizability by word problems of finitely generated semigroups

We first review some old notation, and introduce some new one. If X is a set then the symbol X^* denotes the set of words of elements of X, while with $X^{\mathbb{N}}$ we denote the infinite sequences of elements of X; the symbol $|\sigma|$ for a string σ denotes the length of σ; if $i < |\sigma|$ then $\sigma(i)$ denotes the i-th projection of σ; finally λ denotes the empty string.

Next, we review the notion of Π^0_1 class of the Cantor space, which is central for this section.

Definition 3.5.3. A subset $\mathcal{A}$ of the Cantor space $2^{\mathbb{N}}$ is called a Π^0_1 *class* if $\mathcal{A}$ has a Π^0_1 *definition*, i.e. is of the form $\mathcal{A} = \{X \in 2^{\mathbb{N}} : (\forall n)R(X, n)\}$, for some decidable predicate $R \subseteq 2^{\mathbb{N}} \times \mathbb{N}$. (The Π^0_1 classes are also known as the *effectively closed* subsets of the Cantor space; a subset $\mathcal{A} \subseteq 2^{\mathbb{N}}$ is a Π^0_1 class if and only if $\mathcal{A}$ coincides with the collection of the infinite paths of some decidable tree.)

The reader is referred to Soare's textbook [89] for all unexplained notions regarding computability theory and Π^0_1 classes. Throughout this section when we talk about "degrees" without any further specification, we will mean "Turing degrees".

Finally, if R is an equivalence relation on $\mathbb{N}$, let us denote

$$\mathrm{Tr}(R) = \{T \in 2^{\mathbb{N}} : T \text{ is a transversal of } R\}.$$

Lemma 3.5.4. *For every ceer R,* $\mathrm{Tr}(R)$ *is a* Π^0_1 *class of the Cantor space.*

Proof. The claim follows from the observation that if R is a ceer then

$$T \in \mathrm{Tr}(R) \Leftrightarrow (\forall\, x, y)[x, y \in X \,\&\, x \neq y \Rightarrow x \not R y],$$

which is a Π^0_1 description as the complement of R is co-c.e. $\square$

Throughout the rest of the section, the letter S will be used as a variable for an infinite c.e. semigroup, presented by some given c.e. presentation having word problem $=_S$, using the same conventions introduced in section 3.2.2. Lemma 3.5.6 below isolates a special property of the class of transversal of the ceer $=_S$ when S is finitely generated.

The following observation is an easy rephrasing of Corollary 3.4.4.

Corollary 3.5.5. *If S is an infinite c.e. finitely generated semigroup with X as finite set of generators then*

$$\forall m > 0\, \exists x \in X^* \,[|x| = m \,\&\, (\forall y \in X^*\,)[|y| < m \Rightarrow x \neq_S y]].$$

Lemma 3.5.6. *If S is an infinite c.e. finitely generated semigroup with X as finite set of generators then there exists a nonempty* Π^0_1 *class* $\mathcal{A} \subseteq \mathrm{Tr}(=_S)$ *of the Cantor space whose members are all infinite.*

Proof. Let S and X be as in the statement of the lemma. Identify X^* with $\mathbb{N}$ via a suitable Gödel numbering, and for every $m \in \mathbb{N}$, let X^m denote the strings on alphabet X having length m: clearly $(X^m)_{m\in\mathbb{N}}$ is a strong disjoint array. Define

$$\mathcal{A} = \mathrm{Tr}(=_S) \cap \{A \in 2^{X^*} : (\forall m\,)[A \cap X^m \neq \emptyset]\}.$$

Via identification $X^* = \mathbb{N}$ and thus $2^{X^*} = 2^{\mathbb{N}}$, hence the second class in the previous intersection is a Π^0_1 class: since $\mathrm{Tr}(=_S)$ is Π^0_1 too by Lemma 3.5.4, we have that $\mathcal{A}$ is Π^0_1. Since they are intersected by a disjoint array, all members of $\mathcal{A}$ are infinite. To see that $\mathcal{A} \neq \emptyset$ we build a set $A \in \mathcal{A}$ by specifying by induction on m a string $x \in A$ with $|x| = m$. Let $\lambda \in A$; having specified the strings to place in A of length $< m$, use Fact 3.5.5 to pick and place in A the least string $x \in X^*$ such that $|x| = m$ and $x \neq_S y$ for all $y \in X^*$ such that $|y| < m$. $\square$

In computability theory, Π^0_1 classes are well-studied objects: a number of theorems, known as *basis theorems for non-empty Π^0_1 classes*, concerns degrees of members of *any* such class. For the proofs of the following well-known basis theorems, see e.g. [51].

Fact 3.5.7. *Let $\mathcal{A}$ be a non-empty Π^0_1 class of the Cantor space. Then:*

(i) (Low Basis Theorem) $\mathcal{A}$ *contains a set of low degree.*

(ii) (Hyperimmune-free Basis Theorem) $\mathcal{A}$ *contains a set of hyperimmune-free degree.*

(iii) (Kreisel-Shoenfield Basis Theorem) $\mathcal{A}$ *contains a set $A <_T \emptyset'$.*

As an immediate corollary, we get the existence of certain transversals of $=_S$.

Corollary 3.5.8. *The class* $\mathrm{Tr}(=_S)$ *contains:*

(i) A low infinite transversal;

(ii) a hyperimmune-free infinite transversal;

(iii) an infinite transversal A such that $A <_T \emptyset'$.

Notice that every infinite ceer R contains a co-c.e. infinite transversal, namely the transversal (called the *principal transversal of* R and denoted by T_R) which is the set comprised of the least elements of the various equivalence classes.

The various items of the previous corollary help us find infinite ceers which are not $\equiv$-realized by the word problem of any infinite c.e. finitely generated semigroup.

Definition 3.5.9. Throughout this definition, R ranges through the infinite ceers. Let:

(a) **non-low** $= \{R :$ R does not contain any low infinite transversal$\}$;

(b) **hdark** $= \{R :$ all infinite transversals of R are hyperimmune$\}$;

(c) **hhdark** $= \{R :$ all infinite transversals of R are hyperhyperimmune$\}$;

(d) **non-incomp** $= \{R :$ if T is an infinite transversal of R then $T \not<_T \emptyset'\}$.

The following statement summarizes the respective relations among the classes defined above.

Proposition 3.5.10. *We have (with $\subset$ denoting proper inclusion)*

$$\mathbf{non\text{-}incomp} \cup \mathbf{hhdark} \subset \mathbf{non\text{-}low} \subset \mathbf{hdark};$$

moreover $\mathbf{non\text{-}incomp} \nsubseteq \mathbf{hhdark}$ *and* $\mathbf{hhdark} \nsubseteq \mathbf{non\text{-}incomp}$.

Proof. Except for $\mathbf{non\text{-}low} \subseteq \mathbf{hdark}$ (which will be proved as Theorem 3.5.15), the various inclusions are straightforward. The inclusion $\mathbf{hhdark} \subseteq \mathbf{non\text{-}low}$ follows from the fact that if A is hyperhyperimmune then $\emptyset' <_T A'$, as shown in [50, Theorem 6.1].

In most cases, counterexamples witnessing proper inclusion can be found by taking suitable unidimensional ceers R_X and recalling the following well know facts of classical computability theory. A set X is *introreducible* if, for every infinite subset $Y \subseteq X$, $X \leq_T Y$. In [32], it has been proven that, given any c.e. set Y, there exists a hypersimple set $X \equiv_T Y$ whose complement X^c is introreducible, namely one can take as X the Dekker deficiency set of Y (see also [89], Exercise 5.3.13). Moreover, it has been proven by Yates ([93]) that no deficiency set can be hyperhypersimple. For instance, if we take X to be the deficiency set of a c.e. set which is neither low nor high, then we can show that $R_X \in \mathbf{non\text{-}low} \smallsetminus (\mathbf{non\text{-}incomp} \cup \mathbf{hhdark})$ as follows: $R_X \in \mathbf{non\text{-}low}$ since every infinite transversal A of R_X coincides (modulo one element) with an infinite subset of X^c: therefore by introreducibility $X^c \leq_T A$, and thus $\emptyset' <_T X' \leq_T A'$; on the other hand $R_X \notin \mathbf{non\text{-}incomp}$ as X^c is a transversal and $X^c <_T \emptyset'$; finally $R_X \notin \mathbf{hhdark}$ by Martin's result [62, Corollary 3.1] stating that a c.e. degree is high if and only if it contains a hyperhypersimple set, hence X^c is not hyperhyperimmune. To show that $\mathbf{non\text{-}incomp} \nsubseteq \mathbf{hhdark}$ take as X the deficiency set of $\emptyset'$. To show that $\mathbf{hhdark} \nsubseteq \mathbf{non\text{-}incomp}$ let X be a hyperhypersimple set with $X <_T \emptyset'$ (such a set exists again by [62, Corollary 3.1]): then $R_X \in \mathbf{hhdark} \smallsetminus \mathbf{non\text{-}incomp}$.

Finally, $\mathbf{hdark} \nsubseteq \mathbf{non\text{-}low}$ is proved as Theorem 3.5.14 below. □

Next, we prove that each of the classes from Definition 3.5.9 is $\equiv$-closed, and in fact form an ideal with respect to computable reducibility $\leq$.

Proposition 3.5.11. *If* $\mathbf{P} \in \{\mathbf{non\text{-}low}, \mathbf{hhdark}, \mathbf{hdark}, \mathbf{non\text{-}incomp}\}$ *then*

(A) membership in $\mathbf{P}$ is $\equiv$-invariant, in fact for all ceers E, R, such that $E \leq R$, we have that $E \in \mathbf{P}$ if $R \in \mathbf{P}$;

(B) the $\equiv$-degrees of the ceers in $\mathbf{P}$ *form an ideal.*

Proof. Let us first prove (A). We show that in each one of the specified cases the class $\mathbf{P}$ corresponds to a property P of sets for which we have: if E, R are ceers with $E \leq R$ and R satisfies P then E satisfies P as well. So, let h be a computable function h witnessing that $E \leq R$.

(a) Let T be a low infinite transversal of E. Then $h[T]$ is an infinite transversal of R. Clearly $h[T]$ is c.e. in T, thus $h[T]$ contains an infinite $\widehat{T} \leq_T T$. But then $\widehat{T}$ is an infinite low transversal of R.

(b) Suppose that E intersects an infinite non-hyperimmune transversal T. Then there is a strong disjoint array $(D_{f(n)})_{n\in\mathbb{N}}$ of finite sets which intersects T. Again, $\widehat{T} = h[T]$ is an infinite transversal of R. In order to remedy to the fact that the sequence of finite sets $(D_{g(n)})_{n\in\mathbb{N}}$, where $D_{g(n)} = h[D_{f(n)}]$, need not be a disjoint array, let us consider a computable function k such that, for every n,

$$D_{k(n)} = D_{g(n)} \smallsetminus \left(\bigcup_{m<n} D_{k(m)} \right).$$

Now, $(D_{k(n)})_{n\in\mathbb{N}}$ is a strong disjoint array, and for every n, $D_{k(n)} \cap \widehat{T} \neq \emptyset$, yielding that $\widehat{T}$ is not hyperimmune.

(c) Similar to (b). Suppose that $(W_{f(n)})_{n\in\mathbb{N}}$ is a weak disjoint array which intersects an infinite non-hyperhyperimune transversal T of E, and this time let g be a computable function such that $W_{g(n)} = W_{h(n)}$. Let r be a computable function of two variables such that

$$W_{r(n,s)} = W_{g(n),s} \smallsetminus \left(\bigcup_{\langle m,t\rangle<\langle n,s\rangle} W_{r(m,t)} \right)$$

(we refer of course to uniform computable approximations $\{W_{g(n),s} : n, s \in \mathbb{N}\}$ to the sets of the array $(W_{g(n)})_{n\in\mathbb{N}}$); finally let k be a computable function such that $W_{k(n)} = \bigcup_s W_{r(n,s)}$. It is now easy to see that the sequence $(W_{k(n)})_{n\in\mathbb{N}}$ is a weak disjoint array which intersects $\widehat{T} = h[T]$.

(d) Similar to item (a).

We now prove (B). It is easy to see that if E, R both have property P then $E \oplus R$ has property P. Assume for instance that $E, R \in \mathbf{hdark}$, but $E \oplus R$ has a transversal T which is intersected by a strong disjoint array $(D_{f(n)})_{n\in\mathbb{N}}$. Without loss of generality, we can assume that T contains infinitely many even numbers. Let g be a computable function such that

$$D_{g(n)} = \left\{u : \ 2u \in D_{f(n)}\right\}.$$

Then $(D_{g(n)})_{n\in\mathbb{N}}$ intersects an infinite transversal of E (namely the set $\widehat{T} = \{x : 2x \in T\}$), so that E cannot be hyperdark.

From this, and from the previously proven item (A), we conclude that that the $\equiv$-degrees of the ceers in $\mathbf{P}$ form an ideal. □

We can now prove the main theorem of this section.

Theorem 3.5.12. *If* $\mathbf{P} \in \{\mathbf{non\text{-}low}, \mathbf{hhdark}, \mathbf{hdark}, \mathbf{non\text{-}incomp}\}$ *and* $R \in \mathbf{P}$ *(in fact, by the inclusion of Proposition 3.5.10 this can be summarized by just taking* $R \in \mathbf{hdark}$*) then* R *is not* $\equiv$*-realized by any finitely generated semigroup.*

Proof. By Corollary 3.5.8 and Proposition 3.5.11(A). □

3.5.3 The hyperdark ceers

The class **hdark** is the largest class, among the ones pointed out in the previous section, having the property that no ceer in the class is c-realized by some finitely generated semigroup. Moreover, it marks the line one has to cross to get an immunity property guaranteeing this property: if we drop from hyperimmunity to immunity then this property gets lost, as is shown by the following remarkable result that there are in fact dark ceers which are c-realized even by finitely generated groups.

Fact 3.5.13 (Myasnikov and Osin [72])**.** *There is a c.e. finitely generated group* G *such that* $=_G$ *is dark.*

Up to this point our examples of ceers in **hdark** have been taken only from the class of unidimensional ceers, which are ceers with both finite equivalence classes (in fact, singletons), and infinite equivalence classes (in fact, at most one infinite class: in our example exactly one infinite class). It is a trivial matter to show that each of the classes of ceers presented in Definition 3.5.9

contains ceers with only infinite equivalence classes. Indeed, if R is a ceer then $R \equiv_c R_\infty$ where R_∞ is the ceer defined by

$$\langle i, x\rangle \; R_\infty \; \langle j, y\rangle \Leftrightarrow i = j \;\&\; x \; R \; y,$$

whose equivalence classes are clearly all infinite.

Less trivial is to find examples of ceers lying in the classes of Definition 3.5.9, in which all the equivalence classes are finite. The rest of the section is devoted to showing that there exist ceers in **hdark** with specified properties, having only *finite* equivalence classes. In the following construction, we use the terminology and conventions of Remark 1.2.4: indeed, the following theorem may be seen as a strengthening of Proposition 1.2.5.

Theorem 3.5.14. *There is a hyperdark ceer R such that all R-equivalence classes are finite and its principal transversal is low. It follows that* **hdark** $\not\subseteq$ **non-low**.

Proof. We define in stages a sequence of uniformy decidable ceers $\{R_s\}_{s\in\mathbb{N}}$, such that $R_0 = \mathrm{Id}$, $R_s \subseteq R_{s+1}$ (more precisely, R_{s+1} is obtained by collapsing finitely many classes of R_s), and $R = \bigcup_{s\in\mathbb{N}} R_s$ satisfies the claim. Therefore the sequence of the principal transversals $(T_{R_s})_{s\in\mathbb{N}}$ is uniformly computable, satisfies $T_{R_0} \supseteq T_{R_1} \supseteq \dots$, and thus yields a co-c.e. approximation to the principal transversal T_R of R. Notice that if x, t are such that for all $y < x$ the equivalence class of y does not change after t, i.e. $[y]_{R_t} = [y]_{R_s}$ for all $s \geq t$, then $T_{R_s} \upharpoonright x = T_R \upharpoonright x$. We want to make R satisfy for every $e \in \mathbb{N}$ the following requirements:

$$\begin{aligned}
\mathcal{P}_e &: \text{if } (D_{\varphi_e(n)})_{n\in\mathbb{N}} \text{ is a strong disjoint array} \\
&\quad\; \text{then it does not intersect any transversal of } R; \\
\mathcal{N}_e &: (\exists^\infty s)\left[\varphi_{e,s}^{T_{R_s}}(e)\downarrow\right] \Rightarrow \varphi_e^{T_R}(e)\downarrow; \\
\mathcal{F}_e &: [e]_R \text{ is finite.}
\end{aligned}$$

Satisfaction of all $\mathcal{P}_e$ clearly ensures that R is hyperdark. Satisfaction of all $\mathcal{N}_e$ will ensure that T_R is low (see also [89, pp. 149-151]). Indeed, to decide whether $e \in (T_R)'$, search for a stage s_0 such that either $\varphi_{e,s}^{T_{R_s}}(e)\downarrow$ for all $s \geq s_0$ (in which case $e \in (T_R)'$), or $\varphi_{e,s}^{T_{R_s}}(e)\uparrow$ for all $s \geq s_0$ (in which case $e \notin (T_R)'$). From the satisfaction of all $\mathcal{N}$-requirements it follows that such an s_0 can be effectively found using oracle $\emptyset'$. We also wanto to keep each

R-equivalence class finite by satisfying all $\mathcal{F}$-requirements, in a way similar to the one used in the proof of Proposition 1.2.5.

We prioritize the requirement as follows: $\mathcal{N}_0 < \mathcal{P}_0 < \mathcal{N}_1 < \mathcal{P}_1 < \cdots$.

The strategy pursued by requirement $\mathcal{N}_e$ will be to preserve a computation $\varphi_{e,s}^{T_{R_s}}(e)$ whenever such a computation shows up. To keep track of the restraint imposed by $\mathcal{N}_e$, we consider the computable sequence $(r_s(e))_{s\in\mathbb{N}}$, where $r_s(e)$ denotes the *use* of the computation $\varphi_{e,s}^{T_{R_s}}(e)$ (that is, the greatest element of the oracle A_s queried during the computation $\varphi_{e,s}^{T_{R_s}}(e)$, if this computation converges, or 0 otherwise; see [89, pp. 53; 57] for more details).

To satisfy requirement $\mathcal{P}_e$ the construction will typically *R-collapse* a certain finite set of numbers into a single R-equivalence class. Finally, in order to satisfy $\mathcal{F}$-requirements, namely to keep each R-equivalence class finite, for every e we will define in stages a parameter $B_{e,s}$, starting with $B_{e,0} = \emptyset$, whereas $B_{e,s}$ contains the elements which have been R-collapsed by the action of requirement $\mathcal{P}_e$ by the end of stage s, and we will prevent all requirements having lower priority than $\mathcal{P}_e$ to modify the R-equivalence classes of these elements. If B_e is not explicitly redefined at $s+1$, then it is understood that $B_{e,s+1} = B_{e,s}$.

Construction. We say that a requirement $\mathcal{P}_e$ *requires attention* at stage $s+1$ if $B_{e,s} = \emptyset$ and there exists u, v such that $u \neq v$, $\varphi_{e,s}(u)\downarrow$, $\varphi_{e,s}(v)\downarrow$ and

$$\min\left(D_{\varphi_e(u)} \cup D_{\varphi_e(v)}\right) > \max\{x : (\exists y)(\exists i < e)[x\ R_s\ y\ \&\ [y \leq r_s(i) \vee y \in B_i]\}$$

(where we understand $\max\emptyset = 0$).

At stage $s+1$, if there is no $\mathcal{P}_e$ with $e \leq s$ which requires attention, then go directly to stage $s+2$. Otherwise, let $\mathcal{P}_e$ with $e \leq s$ be the highest priority requirement currently requiring attention, and choose the pair u, v with least pseudocode by which it requires attention. Let $B_{e,s+1} = D_{\varphi_e(u)} \cup D_{\varphi_e(v)}$, and let R_{s+1} be the equivalence relation generated by $R_s \cup \{(x,y) :\ x, y \in B_{e,s+1}\}$. Say that $\mathcal{P}_e$ *acts at* $s+1$ and go to the next stage.

Verification. It is clear that each requirement $\mathcal{P}_e$ acts at most once, as if it acts at stage s, then for all $t \geq s$ we have $B_{e,t} \neq \emptyset$, hence $\mathcal{P}_e$ never requires attention (nor does it possibly acts) at any later stage. In particular, for each e, $B_e = \lim_{s\to\infty} B_{e,s}$ is well-defined: more precisely, either $B_e = B_{e,s}$ if $\mathcal{P}_e$ acts at some (unique) stage s, or $B_e = \emptyset$ otherwise.

We say that a number x *injures* requirement $\mathcal{N}_e$ at stage $s+1$ if $x \leq r_s(e)$ but x belongs to either of the two sets $D_{\varphi_e(u)}, D_{\varphi_e(v)}$ which are R-collapsed at stage $s+1$ by the action of some $\mathcal{P}_i$. Notice that each $\mathcal{N}_e$ is injured only by finitely many elements, and thus only finitely many times: indeed, we R-collapse an element $x \leq r_s(e)$ only if some requirement $\mathcal{P}_i$ acts, with $i < e$. But each such requirement acts at most once, and therefore it R-collapses only finitely many elements.

We now prove that each requirement $\mathcal{N}_e$ is satisfied. For each e, we have seen that there must be a stage s_0 such that $\mathcal{N}_e$ is not injured at any stage $s > s_0$. If $\varphi_{e,s}^{T_{R_s}}(e)\downarrow$ for $s > s_0$ then for all $t \geq s$ we have that $r_t(e) = r_s(e)$. Therefore $T_{R_s} \upharpoonright r_s(e) = T_R \upharpoonright r_s(e)$, as after stage s we never modify the equivalence class of any element below $r_s(e)$. This shows that $\varphi_e^{T_R}(e)\downarrow$ and hence $\mathcal{N}_e$ is satisfied. By satisfaction of all the $\mathcal{N}$-requirements it follows that T_R is low.

Our next claim is that R is hyperdark, as a consequence of the fact tat every requirement $\mathcal{P}_e$ is satisfied. Notice that if $\mathcal{P}_e$ is ever allowed to act, then just this one action is enough to have that if $\{D_{\varphi_e(n)}\}_{n\in\mathbb{N}}$ is a disjoint strong array, then the array does not intersect any transversal T for R. Indeed, whenever $\mathcal{P}_e$ acts, it R-collapses two disjoint finite sets $D_{\varphi_e(u)}, D_{\varphi_e(v)}$ into a single R-equivalence class. Hence, if an infinite set T is intersected by the array $(D_{\varphi_e(n)})_{n\in\mathbb{N}}$, then there are $t \in T \cap D_{\varphi_e(u)}$ and $t' \in T \cap D_{\varphi_e(v)}$ such that $t \neq t'$ (since the array is disjoint) but $t \mathrel{R} t'$, implying that T cannot be a transversal of R.

It remains to show that, if $(D_{\varphi_e(n)})_{n\in\mathbb{N}}$ is a strong disjoint array (thus, in particular φ_e is total and unbounded), then $\mathcal{P}_e$ eventually acts. By what shown above, assume that s_0 is a stage such that there is no $s > s_0$ and no $i < e$ such that r_i or B_i changes at s. Then either $\mathcal{P}_e$ has already acted at some stage $s \leq s_0$, or there must be a stage $s > s_0$ so that $\varphi_{e,s}$ converges on two suitable elements, as φ_e is unbounded and $(D_{\varphi_e(n)})_{n\in\mathbb{N}}$ is a strong disjoint array. In any case, $\mathcal{P}_e$ eventually acts, hence it is eventually satisfied.

Finally, the verification that each $\mathcal{F}$-requirement is satisfied is the same as in the proof of Proposition 1.2.5. To see that for every x, the equivalence class $[x]_R$ is finite, indeed, just observe that either x is never R-collapsed to other numbers by any $\mathcal{P}$-requirement, and thus $[x]_R$ is a singleton, or there is some i such that x is R-collapsed by $\mathcal{P}_i$ at some stage s. But then after s, the number x can be R-collapsed only by $\mathcal{P}$-requirements $\mathcal{P}_j$ with $j < i$, and thus the equivalence class of x changes only finitely many times, since each $\mathcal{P}_j$ acts at most once. □

On the other hand we have:

Theorem 3.5.15. non-low $\subseteq$ hdark.

Proof. Suppose that R is a ceer containing a transversal A which is not hyperimmune as witnessed by the strong disjoint array $(D_{f(n)})_{n\in\mathbb{N}}$, and consider the class of sets

$$\mathcal{A} = \mathrm{Tr}(R) \cap \{A : (\forall n)[A \cap \{x \in D_{f(n)} \,\&\, x \geq n\} \neq \emptyset]\}.$$

It is easy to see that $\mathcal{A}$ is a Π^0_1 class of the Cantor space, as $\mathcal{A}$ is the intersection of two Π^0_1 classes. Moreover it is clear by the definition that all members of $\mathcal{A}$ must be infinite, and $\mathcal{A} \neq \emptyset$ as $A \in \mathcal{A}$. Therefore, by the Low Basis Theorem, $\mathcal{A}$ contains a low member, that is a infinite low transversal of R. By contrapositive this shows that **non-low** $\subseteq$ **hdark**. $\square$

Remark 3.5.16. It might be worth noticing the following easy consequence of the previous theorem: every co-c.e. set X which is not hyperimmune contains an infinite low subset. Indeed, the unidimensional ceer R_{X^c} is not hyperdark, hence it admits an infinite low transversal, which obviously is, in particular, a subset of X (modulo one element).

We conclude by exhibiting a ceer R having all finite equivalence classes, and such that every transversal of R is of hyperhyperimmune degree. In fact, our next example is built so that every transversal of R computes $\emptyset'$, hence, in particular, it is high. Then every transversal has hyperhyperimmune degree since every high degree is hyperhyperimmune (as shown by Jockusch, see [50]). Clearly $R \in$ **non-incomp**, and thus is one more example of a ceer that cannot be $\equiv$-realized by and finitely generated semigroup.

Theorem 3.5.17. *There is a ceer R such that all R-equivalence classes are finite and, if A is a transversal of R, then $\emptyset' \leq_T A$.*

Proof. Given an infinite set A, we denote by p_A the *principal function* of A, namely the function which enumerates the elements of A in strict order of magnitude. It is clear that, if T is a transversal of R, then $p_{T_R}(n) \leq p_T(n)$ (where we recall that T_R denotes the principal transversal of R). We will then make use of the well-known result (proved by Martin [63], and independently by Tennenbaum [90]) that a sufficient condition for $\emptyset' \leq_T A$ is the existence of a function $g \leq_T A$ which dominates every partial computable function, i.e. for every e there exists e_0 such that for every $i \geq e_0$, if $\varphi_e(i)\downarrow$ then $\varphi_e(i) < g(i)$. Hence, to complete our task, it is enough to build a ceer R

with only finite equivalence classes and such that the function $p_{T_R}(n+1)$ dominates every partial computable function. This would show, as argued at the end of the proof, that for every infinite transversal T the function $g(n) = p_T(n+1)$ dominates all partial computable functions, and clearly $g \leq_T T$.

Without loss of generality, we assume that for every e, i, s, if $\varphi_{e,s}(i) \downarrow$ then $\varphi_{e,s}(i) < s$. For every e, s, let

$$f_s(e) = \max\{0, \varphi_{i,s}(j) \downarrow : i, j \leq e\}.$$

For all e, s, it holds that $f_s(e) \leq f_s(e+1)$ and $f_s(e) \leq f_{s+1}(e)$. Moreover, for every e there is a stage s_0 so that, for every $s \geq s_0$, $f_s(e) = f_{s_0}(e)$. Hence, $f(e) = \lim_{s\to\infty} f_s(e)$ is well-defined for every e. To achieve our goal, for every e it is enough to satisfy the requirement

$$\mathcal{R}_e : f(e) < p_{T_R}(e+1),$$
$$\mathcal{S}_e : [e]_R \text{ is finite.}$$

Notice that if $f(e) < p_{T_R}(e+1)$ for every e, then for every $i \geq e$, if $\varphi_e(i) \downarrow$ then $\varphi_e(i) < p_{T_R}(i+1)$. For the $\mathcal{R}$-requirements, consider the priority ordering $\mathcal{R}_i < \mathcal{R}_j$ if $i < j$. Moreover, notice that, in the following construction, we satisfy the $\mathcal{S}$-requirements directly by letting R induce a partition of $\mathbb{N}$ into consecutive finite intervals.

Construction. We define R in stages. At each stage s, our uniformly computable approximation R_s to R will be an equivalence relation partitioning $\mathbb{N}$ in consecutive closed finite intervals $\{r_{e,s} : e \in \mathbb{N}\}$, almost all of which will be singletons, in such a way that $R_s \subseteq R_{s+1}$. Let $t_{e,s} = \max r_{e,s}$: then $t_{e,s} + 1 = \min r_{e+1,s}$.

At stage 0 we start up with $r_{e,0} = \{e\}$, thus $t_{e,0} = e$; thus $R_0 = \mathrm{Id}$.

We say that a requirement $\mathcal{R}_e$ *requires attention* at stage $s+1$ if $f_{s+1}(e) > t_{e,s}$. By our assumption on how to approximate the partial computable functions, we may suppose that $f_{s+1}(e) < s+1$.

At stage $s+1$, see if there is a requirement $\mathcal{R}_e$ with $e \leq s$ which requires attention: if not, then go to stage $s+2$, leaving unchanged each r_i. Otherwise let $\mathcal{R}_e$ be the highest priority requirement requiring attention. Define (assuming $t_{-1} = -1$)

$$r_{j,s+1} = \begin{cases} r_{j,s}, & \text{if } j < e, \\ [t_{e-1,s}+1, s+1], & \text{if } j = e, \\ \{s+1+j-e\}, & \text{if } j > e. \end{cases}$$

This yields R_{s+1} as well. It is easy to see that $R_s \subseteq R_{s+1}$. We say in this case that $\mathcal{R}_e$ *acts*. Go to the next stage.

Verification. The construction is a standard priority argument with finite injury. Notice that, if $\mathcal{R}_e$ acts at a stage s_0 such that $f_{s_0}(e) = f(e)$, then it will never require attention again at a later stage. A straightforward argument by induction on the priority of the requirements shows that for every e the requirement $\mathcal{R}_e$ eventually stops requiring attention, r_e and its maximum t_e reach their limits, with $r_e = [t_{e-1}+1, t_e]$ (where again we assume that $t_{-1} = -1$), and $f(e) \leq t_e$, so that $f(e) < t_e + 1 = p_{T_R}(e+1)$, and the requirement $\mathcal{R}_e$ is satisfied. Moreover, this trivially implies that each $\mathcal{S}$-requirement is satisfied, namely that all R-equivalence classes are finite. Notice that

$$T_R = \{t_{e-1} + 1 : e \in \mathbb{N}\}.$$

Indeed, let s_0 be the least stage such that for all $i < e$, we have that $\mathcal{R}_i$ does not receive attention, and r_i does not change at any later stage. If $\mathcal{R}_e$ never requires attention at any $s+1 > s_0$ then $r_e = r_{e,s_0}$ and $f(e) \leq t_e < t_e + 1 = p_{T_R}(e+1)$. On the other hand, $\mathcal{R}_e$ may require attentions only finitely many times at stages $s+1 > s_0$ as $f_s(e)$ may change only finitely many times: if $s+1 > s_0$ is the last stage such that $\mathcal{R}_e$ requires attention and $\mathcal{R}_e$ acts at $s+1$, then we have that $t_e = s+1$ and for every $u \geq s+1$, $f_u(e) \leq t_e$, thus again $f(e) \leq t_e < t_e + 1 = p_{T_R}(e+1)$.

It trivially follows from this that the function $p_{T_R}(e+1)$ dominates all partial computable functions. It remains to show that if T is any transversal then the principal function p_T dominates all partial recursive functions. Given n, let a_n be such that $p_{T_R}(a_n)$ is the least element in $[p_T(n)]_R$, so that $p_{T_R}(a_n) \leq p_T(n)$. Since the equivalence classes of R are consecutive closed intervals, we have $n \leq a_n$, whence $p_{T_R}(n) \leq p_{T_R}(a_n) \leq p_T(n)$. Therefore $p_T(n+1)$ dominates all partial computable functions because so does $p_{T_R}(n+1)$. $\square$

3.6 Classes of algebras $\simeq_{\mathrm{s}}$-realizing provable equivalence of Peano Arithmetic

Although by Fact 3.4.1 there are ceers R such that $R \not\leq =_G$, for every c.e. group G, it is known that there are f.p. groups G such that $=_G$ is universal. This was first proved by Miller III [68]. Another example, due to [76] refers to the computability theoretic notion of effective inseparability. We recall that a disjoint pair (U, V) of sets of numbers is *effectively inseparable* (*e.i.*) if there exists a partial computable function ψ such that for each pair (u, v), if $U \subseteq W_u$ and $V \subseteq W_v$ and $W_u \cap W_v = \emptyset$ then $\psi(u, v)$ converges and $\psi(u, v) \notin W_u \cup W_v$. A f.p. group G is built in [76] such that $=_G$ is *uniformly effectively inseparable* i.e. uniformly in x, y one can find an index of a partial recursive function ψ witnessing that the pair of sets $([x]_{=_G}, [y]_{=_G})$ is e.i., if $[x]_{=_G} \cap [y]_{=_G} = \emptyset$. Such a f.p. group has universal word problem, since it is known ([3]) that every uniformly effectively inseparable ceer is universal.

An important $\simeq_{\mathrm{s}}$-type among the universal ceers is given by the $\simeq_{\mathrm{s}}$-type of the relation $\sim_T$ of provable equivalence of any consistent formal system T extending Robinson's systems Q or R (see for instance Smorynski [87] for an introduction to formal systems of arithmetic), i.e. $x \sim_T y$ if (identifying sentences with numbers through a suitable Gödel numbering) $T \vdash x \leftrightarrow y$. For example, let us take T to be Peano Arithmetic.

The question naturally arises as to which algebras $\simeq_{\mathrm{s}}$-realize $\sim_T$. Notice that by Fact 1.3.5, "$\simeq_{\mathrm{s}}$-realizing $\sim_T$" is equivalent to "$\simeq$-realizing $\sim_T$". Here are some initial remarks about this question:

1. As far as we know, the question of whether there are f.p. semigroups, or f.p. groups, having word problems strongly isomorphic to $\sim_T$ is still open.

2. On the other hand, by Theorem 3.2.4 there exist c.e. semigroups whose word problem is strongly isomorphic to $\sim_T$. We do not know if there are c.e. groups $\simeq_{\mathrm{s}}$-realizing $\sim_T$.

3. If one computably identifies with numbers the sentences of our chosen formal system T, and considers the computable operations provided by the connectives $\wedge$, $\vee$, $\neg$, $\perp$, $\top$ (where $\perp$ and $\top$ denote any contradiction and any theorem, respectively), then

$$\langle \mathbb{N}, \wedge, \vee, \neg, \perp, \top, E \rangle$$

(where $x\ E\ y$ if $T \vdash x \leftrightarrow y$) is a positive presentation of the Lindenbaum algebra of the sentences of T, which is therefore a c.e. Boolean algebra. It is known that the word problem of this c.e. Boolean algebra is strongly isomorphic to $\sim_T$: see [80] (see also [70]).

The above item (3) identifies a very special class of rings which $\simeq_{\mathrm{s}}$-realize $\sim_T$, namely Boolean rings, i.e. rings satisfying $x^2 = x$ for all x. Is that all? Can we find non-Boolean rings $\simeq_{\mathrm{s}}$-realizing $\sim_T$? We will identify in the following a c.e. ring R which is neither Boolean nor commutative, such that $=_R \simeq_{\mathrm{s}} \sim_T$.

The $\simeq_{\mathrm{s}}$-type of $\sim_T$ can be characterized through the already given notion of a diagonal function, and the notion of *uniformly finite precompleteness*, originating from [69] (see also [85]) and introduced in Definition 1.4.6 above. We have also seen that, for every ceer S, $S \simeq_{\mathrm{s}} \sim_T$ if and only if S is u.f.p. and possesses a diagonal function (see Fact 1.4.7).

The rest of the section is devoted to seeing that there is a non-commutative and non-Boolean c.e. ring whose word problem is strongly isomorphic to $\sim_T$. The following result is essentially a rephrasing of Theorem 4.1 of [4].

Lemma 3.6.1. *Let A be a c.e. algebra whose type contains two binary operations $+, \cdot$, and two constants $0, 1$ such that $+$ is associative, the pair (U_0, U_1) is e.i., where*

$$U_i = \{x : x =_A i\},$$

and, for every a,

$$a + 0 =_A a, \qquad a \cdot 0 =_A 0, \qquad a \cdot 1 =_A a.$$

Then $=_A$ is a u.f.p. ceer.

Proof. For the convenience of the reader, we recall the argument in [4], adapting it to our context and notations. We look for a computable function $f(D, e, x)$ such that if $\phi_e(x)\downarrow$, and $\phi_e(x) =_A d$ for some $d \in D$ then $f(D, e, x) =_A \phi_e(x)$.

Let p be a productive function for the pair (U_0, U_1): it is well known that we may assume that p is total. Let

$$\{u_{d,D,e,x}, v_{d,D,e,x} : \ D \text{ finite subset of } \mathbb{N},\ d \in D, e, x \in \mathbb{N}\}$$

be a computable set of indices we control by the Recursion Theorem. For a pair $(u_{d,D,e,x}, v_{d,D,e,x})$ in this set let $c_{d,D,e,x} = p(u_{d,D,e,x}, v_{d,D,e,x})$ and $a_{d,D,e,x} = d \cdot c_{d,D,e,x}$. Define

$$f(D, e, x) = \sum_{d \in D} a_{d,D,e,x}.$$

Let us define two c.e. sets $W_{u_{d,D,e,x}}$ and $W_{v_{d,D,e,x}}$ for each $d \in D$, which are computably enumerated as follows. Wait for $\phi_e(x)$ to converge to some y which is $=_A$ to some element in D, and while waiting, we let $W_{u_{d,D,e,x}}$ and $W_{v_{d,D,e,x}}$ enumerate U_0 and U_1, respectively. If we wait forever then for all $d \in D$ we end up with $W_{u_{d,D,e,x}} = U_0$ and $W_{v_{d,D,e,x}} = U_1$. If the wait terminates, let $d_0 \in D$ be the first seen so that $\phi_e(x) =_A d_0$, enumerate also $c_{d_0,D,e,x}$ into $W_{u_{d_0,D,e,x}}$: this ends up with $W_{u_{d_0,D,e,x}} = U_0 \cup \{c_{d_0,D,e,x}\}$ and $W_{v_{d_0,D,e,x}} = U_1$, thus forcing $c_{d_0,D,e,x} =_A 1$ (since it must be that $W_{u_{d_0,D,e,x}} \cap W_{v_{d_0,D,e,x}} \neq \emptyset$, for otherwise $c_{d_0,D,e,x} = p(u_{d_0,D,e,x}, v_{d_0,D,e,x}) \in W_{u_{d_0,D,e,x}} \cup W_{v_{d_0,D,e,x}}$, a contradiction) and thus $a_{d_0,D,e,x} =_A d_0$. For all $d \in D$ with $d \neq d_0$, we let $W_{u_{d,D,e,x}} = U_0$ and $W_{v_{d,D,e,x}} = U_1 \cup \{c_{d,D,e,x}\}$: this forces $c_{d,D,e,x} =_A 0$ and thus $a_{d,D,e,x} =_A 0$ for each such d. Therefore $f(D,e,x) = \sum_{d \in D} a_{d,D,e,x} =_A d_0 =_A \phi_e(x)$. □

In order to prove the existence of a ring with the desired properties, let us first recall the notion of *free ring*. For more details on the following construction see for instance paragraph IV.2 of [30].

Let R be a ring and M be a monoid. The *monoid ring* of M over R, denoted RM, is the set

$$\{\varphi : M \to R : \ \mathrm{supp}(\varphi) \text{ is finite}\},$$

where $\mathrm{supp}(\varphi) = \{m \in M : \ \varphi(m) \neq 0\}$, equipped with the following operations. Given $\varphi, \psi \in RM$, their *sum* is the function $\varphi + \psi : M \to R$ given by

$$(\varphi + \psi)(m) = \varphi(m) + \psi(m),$$

and their *product* is the function $\varphi\psi : M \to R$ given by

$$(\varphi\psi)(m) = \sum_{hk=m} \varphi(h)\psi(k).$$

Remark 3.6.2. Equivalently, as is easily seen, RM is the set of formal sums

$$\sum_{m \in M} r_m m,$$

where $r_m \in R$, $m \in M$ and $r_m = 0$ for all but finitely many m, equipped with coefficient-wise sum, and product in which the elements of R commute with the elements of M.

Definition 3.6.3. The *free ring* on a set X (denoted $\mathbb{Z}X^*$) is the monoid ring of the free monoid X^* over the ring $\mathbb{Z}$ of the integers.

Theorem 3.6.4. *There exist non-commutative (and hence non-Boolean) c.e. rings R satisfying that $=_R \simeq_{\mathrm{s}} \sim_T$.*

Proof. Assume that $X = \{x_i : i \in \mathbb{N}\}$ is a decidable set and consider the free ring $R^- = \mathbb{Z}X^*$. Notice that, up to coding, we can identify the universe of R^- with $\mathbb{N}$ and assume that its operations are computable and equality is decidable.

Let $U, V \subseteq \mathbb{N}$ be an e.i. pair of c.e. sets, and consider the ideal K of $\mathbb{Z}X^*$ generated by

$$\{x_i : i \in U\} \cup \{1 - x_j : j \in V\}.$$

Thus, any element of K is of the form

$$\sum_{i \in I} r_i \tau_i x_i^U \rho_i + \sum_{j \in J} s_j \mu_j (1 - x_j^V) \nu_j. \tag{$\dagger$}$$

where each of r_i, s_j is in $\mathbb{Z}$, and each of $\tau_i, \rho_i,\ \mu_j, \nu_j$ is in X^*, and finally $I \subseteq U$ and $J \subseteq V$ are finite sets. Up to shrinking the sets of indices, we can suppose that no further simplification can be made in either sum.

The ideal K gives rise to a congruence, which we still denote with K, such that $[0]_K = K$. We claim that $1 \notin K$, which implies

$$[0]_K \cap [1]_K = \emptyset.$$

To see that our claim is true, we show in fact that no nonzero integer can be written as in ($\dagger$). Calculating we get

$$\sum_{i \in I} r_i \tau_i x_i^U \rho_i + \sum_{j \in J_0} s_j \mu_j \nu_j - \sum_{j \in J_0} s_j \mu_j x_j^V \nu_j + \sum_{j \in J_1} s_j - \sum_{j \in J_1} s_j x_j^V, \tag{$\dagger\dagger$}$$

where $J_0 = \{j \in J : \mu_j \nu_j \neq \lambda\}$ and $J_1 = \{j \in J : \mu_j \nu_j = \lambda\}$. By our assumptions, neither the first, nor the third, nor the last sum of ($\dagger\dagger$) contain any pair of like monomials, so that in these sums no further simplification can be made. In order to get a nonzero integer s from this sum we must have that

$$0 = \sum_{i \in I} r_i \tau_i x_i^U \rho_i + \sum_{j \in J_0} s_j \mu_j \nu_j - \sum_{j \in J_0} s_j \mu_j x_j^V \nu_j - \sum_{j \in J_1} s_j x_j^V, \tag{$\ddagger$}$$

$\sum_{j \in J_1} s_j = s$, and $J_1 \neq \emptyset$.

We are going to see that the assumption $J_1 \neq \emptyset$ leads to a contradiction, by showing that there would be an infinite sequence $\alpha_n = s\sigma_n$ $(n \geq 1)$, with $\sigma_n \in \{x_j : j \in V\}^*$ of length n, and $s \in \mathbb{Z} \smallsetminus \{0\}$, such that each α_n occurs as an summand in the second sum of (‡).

Take $j \in J_0$ and let $s = s_j$. So $-s_j x_j^V$ occurs in the fourth sum of (‡). To cancel the monomial $s_j x_j^V$ in (‡), there must be a monomial of the form $s\mu_{j_1}\nu_{j_1}$ (hence from the second sum) such that $\mu_{j_1}\nu_{j_1} = x_j^V$. Let $\sigma_1 = x_j^V$, and $\alpha_1 = s\sigma_1$. So α_1 satisfies the claim. Now suppose that we have found already $\alpha_n = s\sigma_n$ in the second sum and satisfying the claim. Then σ_n is of the form $\mu_{j_n}\nu_{j_n}$ which (via multiplication $s\mu_{j_n}(1 - x_{j_n})\nu_{j_n}$ in (†)) corresponds to an summand in the third sum $-s\mu_{j_n}x_{j_n}\nu_{j_n}$, so that $\sigma_{n+1} = \mu_{j_n}x_{j_n}\nu_{j_n}$ has length $n+1$, and lies in $\{x_j : j \in V\}^*$. Again, this cannot cancel with anything in the first sum, for each summand in the first sum contains an element indexed from U; it cannot cancel with anything in the fourth sum, nor can it cancel with anything in the third sum, because we have assumed that it does not contain like monomials; so it must cancel with something in the second sum, which therefore contains $\alpha_{n+1} = s\sigma_{n+1}$ satisfying the claim.

Lemma 3.6.5. $(U, V) \leq_m ([0]_K, [1]_K)$*, hence the pair* $([0]_K, [1]_K)$ *is e.i.*

Proof. We want to show that $(U, V) \leq_m ([0]_K, [1]_K)$ via $f(i) = x_i$. Thus we must verify that

$$x_i \in [0]_K \Leftrightarrow i \in U,$$

and

$$x_j \in [1]_K \Leftrightarrow j \in V.$$

The facts that $i \in U$ implies $x_i \in [0]_K$ and $j \in V$ implies $x_j \in [1]_K$ are obvious.

On the other hand, if $x_i \in [0]_K$, then x_i must be of the form (†), from which we obtain again the expression (††), with the same assumptions on already done simplifications. Assume that there is an $x_i \in [0]_k$ with $i \notin U$. Since no nonzero integer must appear, either $J_1 = \emptyset$ or J_1 has at least two elements. Assume the latter. Then in the last sum there is a monomial $-s_j x_j^V$ which must cancel with a like monomial, which can be nowhere but in the second sum. But the existence of such a monomial implies that there is a monomial of the form $s_j x_j^V x_{j'}^V$ or $s_j x_{j'}^V x^V - j$ which in turn leads to a contradiction, by an argument similar to the one above. Thus J_1 must be empty. Now assume J_0 is non-empty, so that there is $j \in J_0$ with $\mu_j \nu_j = x_i$, where $i \notin U$. But then in the third sum there must be a corresponding monomial

$(-x_i x_j^V$ or $-x_j^V x_i)$, whose existence, by reasoning as in the argument used to see that non nonzero integer lies in K, leads again to a contradiction.

Since $x_j \in [1]_K$ if and only if $1 - x_j \in [0]_K$, a completely similar argument shows that $x_j \in [1]_K$ implies $j \in V$.

Effective inseparability of the pair $([0]_K, [1]_K)$ follows from the fact that (U, V) is e.i., and effective inseparability is a $\leq_m$-upwards closed property. □

Consider the ring R obtained by dividing R^- by the congruence K. R is a c.e. ring according to Definition 3.1.1, as it can be positively presented as $\langle \mathbb{N}, F, E \rangle$ where we effectively identify modulo coding R^- with $\mathbb{N}$, F is the set of computable operations on $\mathbb{N}$ which correspond via coding to the operations of R^-, and E is the ceer induced on $\mathbb{N}$ by the congruence K.

Moreover, R is equipped with two binary operations $+, \cdot$ (which are its ring binary operations) and two constants $0, 1$ (again, its ring zero-ary operations). Therefore $=_R$ is a u.f.p. ceer by Lemma 3.6.1. To conclude that $=_R$ is strongly isomorphic to $\sim_T$ is then enough by Fact 1.4.7 that we find a diagonal function for $=_R$. For this, just take any $v \neq_R 0$, and consider the function $d(u) = u + v$. It immediately follows that $d(u) \neq_R u$, for otherwise $v =_R 0$. □

3.7 Conclusion

In this chapter we have investigated the problem of which ceers can be realized as word problems of which algebraic structures.

We have given examples of algebraic properties preventing structures having these properties to realize certain classes of ceers: for instance, we observed that non-periodic semigroups (and, similarly, non-torsion monoids) cannot $\equiv$-realize dark ceers. On the other hand, we have shown the existence of a non-commutative ring whose word problem is in the same strong isomorphism type as $\sim_{\mathsf{PA}}$, the provable equivalence in Peano Arithmetic, hence showing that commutativity (and, in particular, the property of being Boolean) is not necessary for a ring to $\simeq_{\mathsf{s}}$-realize $\sim_{\mathsf{PA}}$.

Furthermore, motivated by the observation that the class of groups is not $\equiv$-complete (i.e. not every ceer can be $\equiv$-realized by the word problem of a c.e. group), we have shown that semigroups and monoids are not only $\equiv$-complete, but even $\simeq$-complete. On the other hand, answering a question by Gao and Gerdes in [45], we have shown that $\equiv$-completeness does not hold for finitely generated semigroups. More precisely, we have pointed out an immunity property, namely hyperdarkness, which guarantees that no ceer sharing this property can be $\equiv$-realized by some finitely generated semigroup:notice that, in contrast, there are dark ceers that can be $\equiv$-realized even by finitely generated groups, as shown in [72].

Part II

Contributions to the theory of logical depth and algorithmic randomness

Introduction

This second part of the thesis collects several results in the field of algorithmic randomness. The aim of algorithmic randomness is to give a satisfactory formalization to the intuitive notion of random individual objects. Our main references for algorithmic randomness are [37] and [75].

Chapter 4 investigate the relativization of the so-called *logical depth* (or, simply, *depth*), introduced by Bennett in [9]. Bennett's notion of depth is usually considered to describe, roughly speaking, the usefulness and internal organization of the information encoded into an object such as an infinite binary sequence, which as usual will be identified with a set of natural numbers: in this context, a set is said to be *deep* if, for any given computable time bound t, the difference between the length of the shortest description of the prefix of length n that can be decoded in time $t(n)$ and the length of its "true" optimal description goes to infinity. In Section 4.1, we review some terminology and known facts from algorithmic randomness which will be useful throughout the rest of the thesis.

We next consider two possible relativizations of the notion of depth to an oracle A. For the first one, which we have called simply A-*depth*, we compare, again for any computable time bound t, the shortest descriptions relative to A that can be decoded within the time bound t against its "true" optimal descriptions relative to A. For the second one, we do the same, but we consider not only computable time bounds, but all A-computable time bounds. For this reason, we refer to this notion as A-*Turing-depth*. While both these notions are quite interesting from the mathematical point of view, the first one certainly is, from a philosophical perspective, the most natural one, as it sticks to computable time bound: talking about usefulness of information, we want to consider only "fast" time bounds, which are reasonably identified with computable time bounds, while A-time bounds can be unfeasible even when having access to A.

In Section 4.2, we consider the first kind of relativization, namely A-depth, and we investigate for various kinds of oracles A whether and how the unrelativized and the relativized version of depth differ. It turns out that the classes of deep sets and of sets that are deep relative to the halting set $\emptyset'$ are incomparable with respect to set-theoretical inclusion. On the other hand, the class of deep sets is strictly contained in the class of sets that are deep relative to any given Martin-Löf-random oracle. The set built in the proof of the latter result can also be used to give a short proof of the known fact that every PA-complete degree is Turing-equivalent to the join of two Martin-Löf-random sets. In fact, we slightly strengthen this result by showing that every DNC_2 function is truth-table-equivalent to the join of two Martin-Löf random sets. Furthermore, we observe that the class of deep sets relative to any given K-trivial oracle either is the same as or is strictly contained in the class of deep sets. Obviously, the former case applies to computable oracles. We leave it as an open problem which of the two possibilities can occur for noncomputable K-trivial oracles. The results presented in this section are collected in [13].

Finally, in Section 4.3, we consider the "full" relativization, that is A-T-depth. We prove the basic properties of this notion, and observe that A-T-depth and A-depth certainly coincide for any A of hyperimmune-free degree, but the class of A-T-deep sets is strictly contained in the one of A-deep sets whenever A is high.

In Chapter 5, we propose a model of probabilistically computable forecasting scheme using the toolkit of algorithmic randomness.

One way to formalize randomness for infinite binary sequences is via the unpredictability approach: namely, we fix a certain class of gambling strategies (which are called *martingales* in this context) that bet on the values of each bit of the sequences and are fairly rewarded when right, and we consider a sequence X random (with respect to the given class) if no such gambling strategy becomes arbitrarily rich while betting on the bits of X. We will talk of *computable randomness* when we allow only total computable martingales, and of *partial computable randomness* if we also consider partial computable martingales.

Notice that, in both cases, we consider *deterministic* martingales. But what does it happen if we allow *probabilistically* computable martingales, too? To answer this question, we have considered a new randomness notion: we call a sequence X *almost everywhere (a.e.) computably random* if, for almost every sequence Y, X is computably random relative to Y (this is indeed equivalent to consider probabilistically computable martingales, as we can assume that Y has been drawn at random by our gamblers in advance). We have then built a partial computable random sequence which is not a.e. computably random, hence proving that probabilistic martingales are actually stronger than deterministic ones. The results of this section have been published in [14].

Chapter 4

Relativization of Bennett's notion of depth

The notion of depth introduced by Bennett can be seen as a formalization of the idea that the same information can be organized in different ways, making certain encodings more or less useful for certain computational purposes. In particular, depth goes beyond just measuring the information encoded into a finite object by its Kolmogorov complexity, i.e., by the length of an optimal effective description of the object as a binary string.

Computability theory provides a paradigmatic example of organizing the same information in different ways by the halting set $\emptyset'$ and Chaitin's Ω. Since $\emptyset'$ is a c.e. set, describing $\emptyset' \upharpoonright n$, the prefix of its characteristic sequence of length n, requires not more than $\mathrm{O}(\log n)$ bits. On the other hand, Chaitin [27] demonstrated that Ω is ML-random, so describing $\Omega \upharpoonright n$ requires approximately n bits. It is well-known that $\Omega \equiv_T \emptyset'$, that is, Ω and $\emptyset'$ can be mutually computed from each other, and in this sense the two sets encode the same information. More specifically, Ω is a compressed version of $\emptyset'$ since the first $\mathrm{O}(\log n)$ bits of Ω are sufficient to decide effectively whether $\varphi_n(n)$ halts, i.e., whether n is in $\emptyset'$. On the other hand, computing $\emptyset'$ from Ω must necessarily be "slow" as one can show that $\emptyset'$ is not truth-table reducible to Ω, i.e., $\emptyset'$ cannot be computed from Ω by an oracle Turing machine that runs within some computable time bound. In fact, $\emptyset'$ is not truth-table reducible to any ML-random set. Direct proofs of the statements above can be found in [26]. Some of these results are also shown below, using a different approach related to depth.

This situation is captured quite well within the framework of depth.

Depth has been introduced by Bennett [9] in order to distinguish "useful" or "organized" information from other information such as random noise. In particular, a set is said to be deep if, for any given computable time bound t, the difference between the length of the shortest description of the prefix of length n that can be decoded in time $t(n)$ and the length of its "true" optimal description goes to infinity. In other words, no prefix of a deep set can be optimally described within any computable time bound. It turns out that neither ML-random nor computable sets are deep, whereas the halting problem $\emptyset'$ is deep. Moreover, by a key property of depth, known in literature as "Slow Growth Law", no deep set is truth-table reducible to a nondeep set, in other words, no deep set can be computed from a nondeep set in a "fast" way.

Depth has received renewed attention by several authors in the last decade, with different goals, such as making precise the interplay between depth and computational strength [79], or finding other natural examples of deep sets other than $\emptyset'$ [17]. Moreover, witnessing the profundity of Bennett's intuition, several variants of the notion of depth have been proposed in the literature, both in computability theory and complexity theory (e.g. [36], [57], [71], [79]), for different purposes.

In this chapter, we consider two ways of relativizing the notion of depth, with the aim of better understanding how an oracle may help in organizing information.

Section 4.1 is devoted to review central definitions and known facts in algorithmic randomness, as well as to present the main known results concerning Bennett's depth.

The core of the chapter is section 4.2, where we investigate a natural relativization of depth. The relativization of depth gains additional interest since it differs in the following respect from most other relativizations considered in computability theory. Usually, when relativizing a class, trivially for all oracles the relativized class contains the unrelativized class or the other way round, i.e., for all oracles the unrelativized class contains the relativized one. Examples are given by the classes of computable and of Martin-Löf random sets, respectively. For the class of deep sets the situation is different since depth is defined in terms of the difference between time-bounded and unbounded Kolmgorov complexity. With access to any given oracle, trivially each individual value of the two latter quantities stays the same or decreases but a priori for any given argument the two corresponding values may decrease by different amounts, hence their difference may increase or decrease.

As a consequence, a priori none of the following four cases can be ruled for a given oracle when comparing the classes of deep sets and of sets that are deep relative to the oracle: first, the two classes may be incomparable with respect to set-theoretical inclusion, second, the unrelativized class may be strictly contained in the relativized class, third, the relativized class may be strictly contained in the unrelativized class and, fourth, the two classes may be the same. We prove that the first case applies to the oracle $\emptyset'$, while the second case applies to all ML-random oracles. As a byproduct of our proof, we slightly strengthen a result due to Barmpalias, Lewis and Ng [7], which states that every PA-complete degree is the join of two ML-random degrees: in fact, we show that every DNC_2 function is truth-table-equivalent to the join of two Martin-Löf random sets. Finally, we observe that every K-trivial oracle falls under the third or fourth case, while case four holds for all computable oracles. We leave it as an open problem whether the third case holds for some or all noncomputable K-trivial oracles. The results presented in this section are included in [13]. This relativization is designed in order to keep focusing on the same class of "fast" computations in so far as it is defined in terms of computable time bounds and not in terms of time bounds computable in the oracle.

Finally, in section 4.3, we consider the case in which full access to the oracle is allowed, namely when the time bounds are merely computable in the oracle. We observe that the Slow Growth Law for this notion of depth holds not only for truth-table reducibility, but for a larger class of reductions, which, in fact, can be seen as tt-reductions relative to a given oracle. Moreover, we show that, for every high oracle, this notion of depth is strictly stronger than the one considered in section 4.2.

4.1 Preliminaries

This section is devoted to review known notions from computability theory and algorithmic randomness used throughout the chapter. We first introduce our notation, which is quite standard and follows mostly the textbooks [37], [75] and [88].

The quantifier $\overset{\infty}{\forall}$ is used to mean "for all but finitely many", while $\overset{\infty}{\exists}$ means "there exists infinitely many". For two real-valued functions f, g and a quantifier Q, we write

$$(Qx)[f(x) \leq^+ g(x)] \text{ (respectively, } (Qx)[f(x) \leq^\times g(x)]).$$

to mean that there exists a constant $c > 0$ such that for all x in the range of Q, $f(x) \leq g(x) + c$ (respectively $f(x) \leq c \cdot g(x)$).

We denote by $\mathbf{2}^{<\mathbb{N}}$ the set of all finite binary strings, while $\mathbf{2}^{\mathbb{N}}$ denotes the Cantor space of all infinite binary sequences (that is, the subsets of $\mathbb{N}$). The empty string is denoted by λ. Given a string σ, its length is denoted by $|\sigma|$. The set of all strings of length n is denoted by $\mathbf{2}^n$. Given a set $X \in \mathbf{2}^{\mathbb{N}}$, we write $X \upharpoonright n$ to denote $X(0)X(1)\dots X(n-1)$, namely the string of the first n bits of X. The same notation for prefixes is used for strings. Moreover, we denote the concatenation of two strings τ and σ by $\tau^\frown\sigma$, or also by $\tau\sigma$.

For a string σ, the *cylinder* $[\sigma]$ is the set of $X \in \mathbf{2}^{\mathbb{N}}$ such that $X \upharpoonright |\sigma| = \sigma$. We work with the product topology on $\mathbf{2}^{\mathbb{N}}$, i.e. the topology generated by all cylinders $[\sigma]$. Moreover, we denote the Lebesgue measure on $\mathbf{2}^{\mathbb{N}}$ by μ, that is, the unique Borel measure such that $\mu([\sigma]) = 2^{-|\sigma|}$ for all σ.

Given a partial computable function φ, where it is understood that φ is computed by some Turing machine which we also denote by φ, and $\sigma \in \mathbf{2}^{<\mathbb{N}}$, we write $\varphi(\sigma)[t]$ to denote the output of $\varphi(\sigma)$ after t steps of computation. Moreover, if $\varphi(\sigma)\downarrow= \rho$, we call σ a *code for ρ (with respect to φ)*.

4.1.1 Kolmogorov complexity

Recall that a set $A \subseteq \mathbf{2}^{<\mathbb{N}}$ is *prefix-free* if no member in A is a prefix of another member of the set. A partial computable function $\varphi : \mathbf{2}^{<\mathbb{N}} \to \mathbf{2}^{<\mathbb{N}}$ is *prefix-free* if its domain is a prefix-free set. For the rest of the paper, we fix a prefix-free machine $\mathcal{U}$ which is *universal* in the sense that for every prefix-free partial computable function φ there exists $\rho_\varphi \in \mathbf{2}^{<\mathbb{N}}$ such that

$$(\forall\sigma)\,[\mathcal{U}(\rho_\varphi\sigma) \cong \varphi(\sigma)]\,,$$

where $\cong$ means that either both sides of the above equation are undefined, or they are both defined and equal. Moreover, we can assume that, if the computation $\varphi(\sigma)$ halts within t steps, then $\mathcal{U}(\rho_\varphi\sigma)$ halts (e.g.) within t^2 steps (as shown in [47]).

A classical approach to measure the information contained in some string is given by its *prefix-free complexity*, which is, roughly speaking, the length of its shortest code with respect to some universal prefix-free machine. We are also interested in *time-bounded* versions of prefix-free complexity. We call a function $t : \mathbb{N} \to \mathbb{N}$ a *time bound* if it is total and non decreasing: then the t-time-bounded Kolmogorov complexity of a string σ is the length of its shortest code running in at most $t(|\sigma|)$ steps.

Definition 4.1.1. The *prefix-free complexity* of $\sigma \in \mathbf{2}^{<\mathbb{N}}$ is

$$K(\sigma) = \min\{|\tau| : \mathcal{U}(\tau)\downarrow = \sigma\}.$$

Given a time bound t, the *t-time-bounded prefix-free complexity* of σ is

$$K^t(\sigma) = \min\{|\tau| : \mathcal{U}(\tau)[t(|\sigma|)]\downarrow = \sigma\}.$$

Note that while the function K only depends on the choice of the universal machine by an additive constant, this is no longer the case for its time-bounded version. However, the assumption on the universal machine that it can simulate any other machine up to a quadratic blow-up in computation time is enough to make all notions presented in this paper independent from the particular choice of universal machine.

We can also equip our universal prefix-free machine $\mathcal{U}$ with some oracle $A \in \mathbf{2}^{\mathbb{N}}$. Given a string σ, its prefix-free complexity relative to A, denoted by $K^A(\sigma)$, is defined by relativizing Definition 4.1.1 in the obvious way. The same applies to its t-time-bounded prefix-free complexity relative to A, which we denote by $K^{A,t}(\sigma)$. Recall that $A \in \mathbf{2}^{\mathbb{N}}$ is *Turing reducible* to $B \in \mathbf{2}^{\mathbb{N}}$, and we write $A \leq_T B$, if there is an oracle machine φ such that $\varphi^B(n) = A(n)$ for all n. Moreover, A is *truth-table reducible* (or *tt-reducible*) to B, and we write $A \leq_{tt} B$, if $A \leq_T B$ via some oracle machine φ such that φ^X is total for every oracle X. Equivalently, $A \leq_{tt} B$ if there is a computable time bound t and an oracle machine φ such that $\varphi^B(n)[t(n)]\downarrow = A(n)$ for every every n (see, e.g., [75, Prop. 1.2.22]). The following lemma shows how reductions among sets relate with their relative strength in compressing strings.

Lemma 4.1.2. *Let $A, B \in \mathbf{2}^{\mathbb{N}}$.*

(i) If $A \leq_T B$, then $K^B(\sigma) \leq^+ K^A(\sigma)$.

(ii) If $A \leq_{tt} B$, then for every computable time bound t there is a computable time bound t' such that $K^{B,t'}(\sigma) \leq^+ K^{A,t}(\sigma)$.

Proof. To prove (i), just observe that any optimal A-code τ for σ is also a B-code for σ, as we may consider a Turing machine which first computes the required bits of A using oracle B and then simulates the computation $\mathcal{U}^A(\tau)$.

Moreover, whenever A is computable from B in some computable time bound (that is, $A \leq_{tt} B$), clearly every A-t-fast-code for σ is also a B-t'-fast-code for σ, for any computable time bound t' which allows to compute the required bits of A from B and to perform the remaining required computation. □

4.1.2 Lower-semicomputable discrete semimeasures

Another way to look at the prefix-free Kolmogorov complexity function, which will be very useful in this paper, is via *lower-semicomputable discrete semimeasures*.

Definition 4.1.3. (i) A *discrete semimeasure* (which we call also simply a *semimeasure*) is a function $m : \mathbf{2}^{<\mathbb{N}} \to [0, \infty)$ such that $\sum_\sigma m(\sigma) \leq 1$. It is *lower-semicomputable* if there is a uniformly computable family of functions $m_s : \mathbf{2}^{<\mathbb{N}} \to \mathbb{Q}$ such that, for any string σ,

$$(\forall s)[m_{s+1}(\sigma) \geq m_s(\sigma)] \quad \text{and} \quad \lim_{s\to\infty} m_s(\sigma) = m(\sigma).$$

We will write *lss* for lower-semicomputable discrete semimeasure.

(ii) A lss m is *universal* if, for each lss m',

$$(\forall \sigma) \left[m'(\sigma) \leq^\times m(\sigma)\right].$$

We recall the following known facts about lss.

Theorem 4.1.4 (Levin, see paragraph 3.9 of [37])**.**
(i) There exists a universal lss, and from now on we fix one of them which we denote by $\mathbf{m}$.
(ii) The function $\sigma \mapsto 2^{-K(\sigma)}$ *is a universal lss. Since two universal lss are, by definition, within a multiplicative constant of one another, it follows that* $K(\sigma) =^+ -\log \mathbf{m}(\sigma)$.

Since $\mathbf{m}$ is lower-semicomputable it can be represented by a non-decreasing family of uniformly computable functions $(\mathbf{m}_s)$. This allows us to define the time-bounded version of $\mathbf{m}$.

Definition 4.1.5. Let $t : \mathbb{N} \to \mathbb{N}$ be a computable time bound. The t-time-bounded version $\mathbf{m}^t$ of $\mathbf{m}$ is the function defined for all σ by

$$\mathbf{m}^t(\sigma) = \mathbf{m}_{t(|\sigma|)}(\sigma)$$

We will constantly make use of the following easy lemma which will allow us to switch between time-bounded Kolmogorov complexity, time-bounded semimeasures and computable semimeasures.

Lemma 4.1.6. *For any given computable time bound t, $\mathbf{m}^t$ is a computable semimeasure. Conversely, if m is a computable semimeasure, there exists a computable time bound t such that $m \leq^\times \mathbf{m}^t$. In particular, for any given time bound t, 2^{-K^t} is a computable semimeasure, hence there exists a computable time bound t' such that $2^{-K^t} \leq^\times \mathbf{m}^{t'}$ (or equivalently, $-\log \mathbf{m}^{t'} \leq^+ K^t$).*

Proof. That $\mathbf{m}^t$ is a computable semimeasure is immediate from the definition. Let now m be a computable semimeasure. It is in particular lower-semicomputable hence there is a constant $c > 0$ such that $m < c \cdot \mathbf{m}$. Since $\mathbf{m} = \lim_s \mathbf{m}_s$, for all n, it suffices to take $t(n)$ to be the smallest s such that $m(\sigma) < c \cdot \mathbf{m}_s(\sigma)$ for all σ of length n. It is immediate that t is computable. The rest of the lemma follows. $\square$

Moreover, we have the following time-bounded version of Theorem 4.1.4: for a proof of the theorem below, see [58, Theorem 7.6.1].

Theorem 4.1.7. *For every computable time bound t, there exists a computable time bound t' such that $\mathbf{m}^t \leq^\times 2^{-K^{t'}}$ (or, equivalently, $K^{t'} \leq^+ -\log \mathbf{m}^t$).*

4.1.3 Martin-Löf randomness

We are often interested in the information encoded into the prefixes of some set. The most studied effective randomness notion for sets, *Martin-Löf randomness*, can be defined in terms of incompressibility of their prefixes (this is also called the "incompressibility" approach to ML-randomness).

Definition 4.1.8. $X \in \mathbf{2}^{\mathbb{N}}$ is *Martin-Löf random* (or simply *ML-random*) if

$$(\forall n)\,[K(X \upharpoonright n) \geq^+ n].$$

One reason for which Martin-Löf's definition of algorithmic randomness is considered to be so important is its robustness, in that one can naturally get this same notion by different approaches. The other two main equivalent approaches to define Martin-Löf randomness are in terms of *ML-tests* and of lower-semicomputable *martingales*.

Definition 4.1.9. (i) A *ML-test* is a sequence of uniformly Σ_1^0 classes $(\mathcal{U}_n)_{n\in\mathbb{N}}$ such that

$$(\forall n)[\mu(\mathcal{U}_n) \leq 2^{-n}].$$

(ii) A *martingale* is a function $d : \mathbf{2}^{<\mathbb{N}} \to [0, \infty)$ such that, for every string σ, $2d(\sigma) = d(\sigma 0) + d(\sigma 1)$.

A ML-test corresponds to an atypical (i.e. having measure 0) property, which can be effectively tested at different levels of confidence: therefore, according to the "stochastic" approach, a ML-random set should be one that withstands all effective statistical tests. On the other hand, a martingale represents the outcome of a gambling strategy in a fair game where debts are not allowed and where the gambler must guess the bits of a sequence one by one, by betting some money at each stage: if the guess is correct, the stake is doubled, lost otherwise. Thus, we can introduce the "unpredictability" approach, according to which we would consider a sequence random whenever its bits cannot be guessed with better-then-average accuracy. We state the well-known equivalence of the three approaches above, first proven by Schnorr [82].

Proposition 4.1.10. *For $X \in \mathbf{2}^{\mathbb{N}}$, the following statements are equivalent.*

(i) X is ML-random.

(ii) For any ML-test $(\mathcal{U}_n)_{n\in\mathbb{N}}$,

$$X \notin \bigcap_{n\in\mathbb{N}} \mathcal{U}_n.$$

(iii) For any lower-semicomputable martingale d,

$$\limsup_{n\to\infty} d(X \upharpoonright n) < \infty.$$

(iv) For any lower-semicomputable martingale d,

$$\liminf_{n\to\infty} d(X \upharpoonright n) < \infty.$$

For a proof, see Theorems 6.3.2 and 6.3.4 in [37], or Theorem 3.2.9 and Proposition 7.2.6 in [75].

It is well-known that there exists a *universal* ML-test, namely a ML-test $(\mathcal{U}_n)_{n\in\mathbb{N}}$ such that if $X \notin \cap_n \mathcal{U}_n$, then X is Martin-Löf random (see Theorem 6.2.5 in [37], or Fact 3.2.4 in [75]). Similarly, there exists a *universal* lower-semicomputable martingale, namely a lower-semicomputable martingale $\mathbf{d}$ such that $\liminf_n \mathbf{d}(X \upharpoonright n) < \infty$ if and only if X is Martin-Löf random (see Corollary 6.3.5 in [37], or Theorem 7.2.8 in [75]).

Finally, we recall that relativized ML-randomness can be equivalently defined by relativizing all three approaches above in the obvious way.

4.1.4 Depth

In many cases, the same information may be organized in different ways, making it more or less useful for various computational purposes. The notion of *depth* was introduced by Bennett in [9] as an attempt to separate useful and organized information from random noise and trivial information.

Definition 4.1.11. $X \in \mathbf{2}^{\mathbb{N}}$ is *deep* if, for every computable time bound $t : \mathbb{N} \to \mathbb{N}$

$$\lim_{n\to\infty} K^t(X \upharpoonright n) - K(X \upharpoonright n) = +\infty.$$

If X is not deep, it is called *shallow*.

By Lemma 4.1.6, we can equivalently define a set X to be deep if and only if, for every computable time bound t,

$$\lim_{n\to\infty} \frac{\mathbf{m}(X \upharpoonright n)}{\mathbf{m}^t(X \upharpoonright n)} = +\infty,$$

or also, if and only if, for every computable semimeasure m,

$$\lim_{n\to\infty} \frac{\mathbf{m}(X \upharpoonright n)}{m(X \upharpoonright n)} = +\infty.$$

Bennett's original definition of depth and the Slow Growth Law

According to Bennett's original definition of depth, a sequence is deep if each code τ of almost every prefix of that sequence which can be decoded in some computable time have optimal description which is arbitrarily shorter than the code τ itself. Formally, we say that a set X is *Bennett-deep* if, for any computable time bound t and $c \in \mathbb{N}$,

$$\left(\overset{\infty}{\forall} n \forall \tau \in \mathcal{U}[t]^{-1}(X \upharpoonright n)\right) [K(\tau) \leq |\tau| - c], \tag{4.1}$$

where $\mathcal{U}[t]^{-1}(\sigma) = \{\tau : \mathcal{U}(\tau)[t(|\sigma|)] \downarrow = \sigma\}$.

The equivalence between Bennett's original definition and Definition 4.1.11, though already stated in [9], has been proven in [53]. In the following, we give a shorter and more direct proof of this equivalence. Notice that, roughly speaking, the equivalence of these definitions says that sequences whose t-fast codes have optimal description arbitrarily shorter than themselves are the same whose optimal description is arbitrarly shorter than any t-fast optimal description.

Theorem 4.1.12. *Let t be a computable time bound. Then:*

(i) If σ is a string such that

$$\left(\forall \tau \in \mathcal{U}[t]^{-1}(\sigma)\right) [K(\tau) \leq |\tau| - c],$$

then $K^t(\sigma) - K(\sigma) \geq^+ c$.

(ii) There is a computable time bound t', depending only on t, such that, if σ is a string with $K^{t'}(\sigma) - K(\sigma) \geq c$, then

$$\left(\forall \tau \in \mathcal{U}[t]^{-1}(\sigma)\right) \left[K(\tau) \leq^+ |\tau| - c\right].$$

In particular, $X \in \mathbf{2}^{\mathbb{N}}$ is deep (according to Definition 4.1.11) if and only if X is Bennett-deep, that is if and only if (4.1) *holds for every computable time bound t and constant c.*

Proof. We start by proving item (i). We first observe that, if τ is a description of σ, then an optimal description of σ must be shorter than an optimal description of τ, up to an additive constant depending only on the underlying fixed universal prefix-free machine.

Claim. *For all* $\tau \in \mathcal{U}^{-1}(\sigma)$, $K(\sigma) \leq^+ K(\tau)$.

Proof of claim. Let $\tau \in \mathcal{U}^{-1}(\sigma)$ and let τ^* be an optimal description of τ. Moreover, consider a prefix-free machine M such that, on input ρ, simulates the computation $\mathcal{U}(\mathcal{U}(\rho))$. Clearly, $\tau^* \in M^{-1}(\sigma)$, hence

$$K(\sigma) \leq^+ K_M(\sigma) \leq^+ |\tau^*| = K(\tau),$$

where the equality holds as τ^* is an optimal description of τ. ■

Now suppose that

$$\left(\forall \tau \in \mathcal{U}[t]^{-1}(\sigma)\right) [K(\tau) \leq |\tau| - c],$$

and let τ be an optimal t-fast description of σ. Then

$$K(\sigma) \leq^+ K(\tau) \leq |\tau| - c = K^t(\sigma) - c,$$

where the first inequality follows by the above claim, while the equality holds as τ is an optimal t-fast description of σ.

We now prove item (ii). Consider the function m defined by

$$m(\sigma) = \sum_{\tau \in \mathcal{U}[t]^{-1}(\sigma)} 2^{-|\tau|}.$$

Since $\mathcal{U}[t]^{-1}(\sigma)$ is always finite and uniformly computable from σ, m is computable. Moreover,

$$\sum_\sigma m(\sigma) = \sum_\sigma \sum_{\tau \in \mathcal{U}[t]^{-1}(\sigma)} 2^{-|\tau|} \leq \sum_{\tau \in \mathrm{dom}(\mathcal{U})} 2^{-|\tau|} \leq 1,$$

meaning that m is a computable discrete semimeasure. By Lemma 4.1.6 and Theorem 4.1.7, there is a computable time bound t' such that $2^{-K^{t'}(\sigma)} \geq^\times m(\sigma)$. Let σ satisfy $K^{t'}(\sigma) - K(\sigma) \geq c$, so that also $\frac{\mathbf{m}(\sigma)}{m(\sigma)} \geq^\times 2^c$ holds. Finally, consider the function m' defined by

$$m'(\tau) = \begin{cases} \frac{2^{-|\tau|}\mathbf{m}(\sigma)}{m(\sigma)}, & \text{if } \tau \in \mathcal{U}[t]^{-1}(\sigma) \\ 0, & \text{otherwise} \end{cases}.$$

Clearly, m' is lower-semicomputable. Moreover,

$$\sum_\tau m'(\tau) = \sum_\sigma \frac{\mathbf{m}(\sigma)}{m(\sigma)} \sum_{\tau \in \mathcal{U}[t]^{-1}(\sigma)} 2^{-|\tau|} = \sum_\sigma \mathbf{m}(\sigma) \leq 1,$$

meaning that m' is a lss. Hence

$$\mathbf{m}(\tau) \geq^\times m'(\tau) = 2^{-|\tau|} \frac{\mathbf{m}(\sigma)}{m(\sigma)} \geq 2^{-|\tau|+c}.$$

Then, the thesis follows by Theorem 4.1.4. $\square$

Bennett's definition is particularly convenient to adopt in order to prove a fundamental property of depth, which is known in literature as the Slow Growth Law and states that depth is upwards-preserved under tt-reductions. Since tt-reductions can be regarded, in some sense, as "fast" oracle computations, the Slow Growth Law may be interpreted as saying that no deep object can be fast computed by shallow objects. We give a simple proof of the Slow Growth Law, which follows the line of the one given in [53].

Proposition 4.1.13 (Slow Growth Law). *Let X be deep and $X \leq_{tt} Y$. Then Y is also deep.*

Proof. Since $X \leq_{tt} Y$, there are total computable functions $\Phi : \mathbf{2}^{<\mathbb{N}} \to \mathbf{2}^{<\mathbb{N}}$ and $f : \mathbb{N} \to \mathbb{N}$ such that $\Phi(Y \restriction n) = X \restriction f(n)$, for every n. Without loss of generality, we can assume that f is non-decreasing and unbounded. Assume that Y is shallow, so that there exist a computable time bound t and a constant c such that, for infinitely many n, there is a t-fast code τ for $Y \restriction n$ with $K(\tau) > |\tau| - c$.

Consider the prefix-free machine M defined as follows. On input ρ, first M simulates $\mathcal{U}(\rho)$. Then, whenever this computation halts, M outputs the string $\Phi(\mathcal{U}(\rho))$. Let τ be a t-fast code for $Y \restriction n$ with $K(\tau) > |\tau| - c$: clearly $M(\tau) = X \restriction f(n)$, hence τ is a t'-fast code for $X \restriction f(n)$, for some computable time bound t', which only depends on t and the tt-reduction Φ. Since we can find such a τ for arbitrary long prefixes of Y, hence for infinitely many prefixes of X, X must be shallow, too. $\square$

Examples of deep and shallow sets

The fact that ML-random sequences must be shallow was already noticed by Bennett in [9], and proven in [53]. We give an easy proof of this fact, along the line of the proof in [53]. As an immediate consequence of this fact and the Slow Growth Law (Proposition 4.1.13), we get, as expected, that also computable sequences are shallow.

Proposition 4.1.14. *(i) Every ML-random set is shallow.*

(ii) Every computable set is shallow.

Proof. We begin by proving item (i). Consider the Turing machine M defined as follows. On input τ, M looks for strings σ and ρ such that $\tau = \sigma\rho$ and $\mathcal{U}(\sigma)$ halts in exactly $|\rho|$ steps. Whenever such strings are found, M outputs $\mathcal{U}(\sigma)\rho$. Notice that M can be assumed to run in some time bound t which is polynomial in the length of the output.

Let X be a ML-random set and $d(n) = K(X \upharpoonright n) - n$. Recall that, as X is ML-random, $\lim_{n\to\infty} d(n) = \infty$ (see, e.g., [75, Proposition 3.2.21]). Hence,

$$(\overset{\infty}{\exists} n)(\forall n' > n)[d(n') > d(n)].$$

For such an n, let σ be an optimal code for $X \upharpoonright n$ and ρ be such that $\mathcal{U}(\sigma)$ halts in exactly $|\rho|$ steps and $(X \upharpoonright n)\rho = X \upharpoonright n'$. Then

$$K^t(X \upharpoonright n') \leq^+ |\sigma|+|\rho| = d(n)+n+|\rho| = d(n)+n' \leq d(n')+n' = K(X \upharpoonright n'),$$

witnessing that X is shallow.

Since, by item (i), there exist shallow sets, item (ii) follows immediately by Proposition 4.1.13. □

The information contained in computable sets is thus regarded as trivial by means of Bennett's depth. A strictly larger class which is typically regarded as computationally weak in algorithmic randomness is that of the so-called *K-trivial* sets, whose definition is recalled below.

Definition 4.1.15. A set $X \in \mathbf{2}^{\mathbb{N}}$ is *K-trivial* if, for all n, $K(X \upharpoonright n) \leq^+ K(n)$.

Hence, K-trivial sets are those whose every prefix has minimal Kolmogorov complexity, thus encoding minimal possible information. The following is a well-known characterization of the K-trivial sets.

Proposition 4.1.16 (Nies, [74])**.** *For a set $X \in \mathbf{2}^{\mathbb{N}}$, the following statements are equivalent.*

(i) X is K-trivial.

(ii) X is low for K*, namely for every string σ, $K(\sigma) \leq^+ K^A(\sigma)$.*

(iii) X *is* low for ML-randomness*, namely every ML-random set is also ML-random relative to* X*.*

In [79], it has been proven that every K-trivial set must be shallow.

Concerning examples of deep sets, Bennett has noticed in [9] that the halting problem $\emptyset'$ is deep. Though a proof of this fact can be found in [53], we give below a more direct proof.

Proposition 4.1.17. *The halting problem* $\emptyset'$ *is deep.*

Proof. Notice that we consider $\emptyset' = \{\langle e, x\rangle : \ \varphi_e(x)\downarrow\}$.

Given any computable time bound t, we construct a Turing machine M. By the Recursion Theorem, we can use an index e for M during the construction.

Let $I_0, I_1, \ldots$ and $J_0, J_1, \ldots$ be partitions of $\mathbb{N}$ into consecutive intervals such that

$$\max I_k = 2^{k+1} \quad \text{and} \quad \max J_k = \left\langle e, 2^{k+1} \right\rangle,$$

hence, for $k > 0$, each interval I_k contains 2^k elements.

Recall that the standard pairing function $\langle\cdot,\cdot\rangle$ has the following properties:

- if $e + i \leq e' + i'$, then $\langle e, i\rangle \leq \langle e', i'\rangle$;
- $\langle e, i\rangle \leq (e + i)^2$.

Thus, in particular, the function $y \mapsto \langle e, y\rangle$ maps elements in I_k to elements in J_k.

Since $\emptyset'$ is c.e., for all $n \in J_{k+1}$ we get

$$K(\emptyset' \upharpoonright n) \leq 4 \log n \leq 4 \log\left(\max J_{k+1}\right) \leq 4 \log(e + 2^{k+2})^2.$$

Let

$$P_k = \left\{p : |p| < 2^k \text{ and } U(p) = w \text{ in at most } t(\max J_{k+1}) \text{ steps where } |w| \in J_{k+1}\right\}.$$

We will define M in such a way that for all $n \in J_{k+1}$

$$K^t(\emptyset' \upharpoonright n) \geq 2^k, \tag{4.2}$$

so that

$$K^t(\emptyset' \upharpoonright n) - K(\emptyset' \upharpoonright n) \geq 2^k - 4 \log(e + 2^{k+2})^2,$$

which is clearly eventually larger than any constant, meaning that H is indeed deep.

To show (4.2), it is enough to ensure that there is no $p \in P_k$ such that $U(p) = \emptyset' \upharpoonright |U(p)|$. To ensure that, we use the 2^k inputs for M in I_k to diagonalize against the at most 2^{2^k} programs in P_k. We define M as follows. On input y, first M computes the index k such that $y \in I_k$ and the sets I_k and P_k. Then, M computes the set

$$P_y = \{p \in P_k : (\forall y' < y,\ y' \in I_k)\,[U(p)\,(\langle e, y'\rangle) = \emptyset'\,(\langle e, y'\rangle)]\}\,.$$

Finally, M halts if and only if for at least half of the programs $p \in P_y$ it holds that

$$U(p)\,(\langle e, y\rangle) = 0.$$

Hence, $M(y)$ diagonalizes against at least half of the programs $p \in P_y$, in the sense that

$$\emptyset'(\langle e, y\rangle) = M(y) \neq U(p)\,(\langle e, y\rangle)\,.$$

Thus, for each $y \in I_k$, we make sure that, for at least half of the remaining programs $p \in P_k$, $U(p) \neq \emptyset' \upharpoonright |U(p)|$ and, since P_k contains at most 2^{2^k} programs and I_k has 2^k many elements, we eventually diagonalize against the whole set P_k. □

A set X is said to be *order-deep* if there exists a computable, non-decreasing and unbounded function g such that, for all computable time bounds t,

$$\left(\overset{\infty}{\forall} n\right)\left[K^t(X \upharpoonright n) - K(X \upharpoonright n) \geq g(n)\right].$$

Notice that the above proof of Proposition 4.1.17 shows, in fact, that $\emptyset'$ is order-deep.

Depth and computational strength

It has been proven in [53] that every high degree contains a deep set. Moreover, in [79], it has been investigated the computational strength of the order-deep sets. In particular, it has been shown that every order-deep set is either high or it computes a *diagonally non-computable* function f (i.e. a function $f : \mathbb{N} \to \mathbb{N}$ with $f(e) \neq \varphi_e(e)$ for all e). As every left-c.e. set which computes a diagonally non-computable function is Turing equivalent to $\emptyset'$, hence is high (as follows easily from Arslanov's Completeness Criterion, [6]), we have in particular that every left-c.e. order-deep set must be high.

We next prove that, in fact, the c.e. degrees containing an order-deep set are exactly the high c.e. degrees. Indeed, by the following theorem, in every such degree, one can construct a c.e. order-deep set.

Theorem 4.1.18. *Let $A \in \mathbf{2}^{\mathbb{N}}$ be high and of c.e. degree. Then there exists a c.e. $B \in \mathbf{2}^{\mathbb{N}}$ which is order-deep and such that $A \equiv_T B$.*

Proof. In the following, by *order*, we mean a non-decreasing, unbounded function $f : \mathbb{N} \to \mathbb{N}$. Since A is high, there exists a function $T \leq_T A$ which dominates any computable function. Moreover, we can assume that T is a left-computable order, namely that we deal with a computable approximation $\{T_s\}_{s\in\mathbb{N}}$ to T such that

$$(\forall s \forall n)\,[T_s(n) \leq T_{s+1}(n) \;\&\; T_s(n) \leq T_s(n+1)]\,.$$

Indeed, let C be a c.e. set such that $A \equiv_T C$ and let $\hat{T} \leq_T C$ a function dominating any computable function. Moreover, let $\{C_s\}_{s\in\mathbb{N}}$ be a computable approximation to C and M be an oracle Turing machine computing $\hat{T}$ according to this approximation, namely

$$(\forall s \forall n)\left[\hat{T}_s(n) = M^{C_s}(n)[s]\right].$$

Then we can define T with the desired properties by letting, for each $s, n \in \mathbb{N}$,

$$T_s(n) = \max_{i \leq s;\; j \leq n} \left\{ M^{C_i}(j)[i] :\; M^{C_i}(j)[i]\downarrow \right\} \cup \{0\}\,.$$

Partition $\mathbb{N}$ into consecutive intervals $I_0, I_1, \ldots$ such that

$$I_0 = \{0,1\}\,, \quad I_m = \left\{2^m, \ldots, 2^{m+1}-1\right\}, \text{ for } m \geq 1.$$

Furthermore, let $\delta < \frac{1}{2}$ and h be any computable order with $h(n) \leq \delta n$ and, for each $m \in \mathbb{N}$, let

$$k_m = 2^{h(m)} \quad \text{and} \quad l_m = 2^{m-h(m)}.$$

Then we partition each interval I_m into k_m consecutive subintervals $I_m^0, \ldots, I_m^{k_m-1}$ of length l_m.

Let $\varphi_0, \varphi_1, \ldots$ be an effective listing of all partial computable functions. For each $m \in \mathbb{N}$, define

$$J_m = \{0 \leq e \leq k_m - 2 : \ \varphi_e(\max I_{m+1})[T_m(m)] \downarrow\}.$$

Moreover, for each $m \in \mathbb{N}$ and $e \in J_m$, let

$$P_m^e = \{p \in \mathbf{2}^{<\mathbb{N}} : |p| \leq l_m \ \& \ \mathcal{U}(p) \downarrow = w \text{ in at most } \varphi_e(\max I_{m+1}) \text{ steps, with } |w| \in I_{m+1}\}.$$

Now we define the set B by specifying the bits of B at each subinterval I_m^e. Namely, for each $m \in \mathbb{N}$ and $0 \leq e \leq k_m - 2$, let us denote

$$B\left[I_m^e\right] = B(2^m + el_m) \ldots B(2^m + (e+1)l_m - 1).$$

Then define each $B\left[I_m^e\right]$ as follows:

- We use the first subintervals to code one bit of A, in order to ensure $B \equiv_T A$, thus define
$$B\left[I_m^0\right] = A(m)0^{l_m - 1}$$
- In case $e \notin J_m$, let $B\left[I_m^{e+1}\right] = 0^{l_m}$.
- Otherwise, $\varphi_e(\max I_{m+1})$ is defined. Thus we use the l_m bits of the intervals I_m^{e+1} to diagonalize against the at most 2^{l_m} possible codes for $B \upharpoonright \max I_{m+1}$ which runs in at most $\varphi_e(\max I_{m+1})$ steps. Namely, for each $y \in I_m^{e+1}$, compute
$$P_y = \left\{p \in P_m^e : \ \left(\forall y' < y \in I_m^{e+1}\right)\left[\mathcal{U}(p)(y') = B(y')\right]\right\}$$
and define $B(y) = 1$ if and only if $\mathcal{U}(p)(y') = 0$ for at least half of the codes $p \in P_y$.

Clearly, the set B constructed in this way is c.e.

Moreover, by construction, for every computable time bound t, we have diagonalized against all possible t-fast codes of length at most l_m. Hence, for every $n \in I_{m+1}$, we have

$$K^t(B \upharpoonright n) > l_m = 2^{m-h(m)}.$$

Note that $2^{m+1} \leq n \leq 2^{m+2} - 1$, thus $\lceil \log n \rceil = m + 2$. For simplicity, define $\log n = \lceil \log n \rceil - 2 = m$. Then we get that, for every computable time bound t and every $n \in \mathbb{N}$,

$$K^t(B \upharpoonright n) > 2^{\log n - h(\log n)}.$$

On the other hand, the bits $B[I_m^e]$, with $e \geq 1$, are uniformly computable, given the information whether $e \in J_m$. Hence, in order to describe $B \upharpoonright n$ it is enough to give the length n and the information whether $e \in J_m$, for each $m \leq \log n$ and each $0 \leq e < 2^{h(m)}$. Thus, we get that, for any $n \in I_{m+1}$,

$$K(B \upharpoonright n) \leq^+ m \cdot 2^{h(m)},$$

so that, for any $n \in \mathbb{N}$,

$$K(B \upharpoonright n) \leq^+ \log n \cdot 2^{h(\log n)}.$$

Hence, we finally get that, for every computable time bound t,

$$K^t(B \upharpoonright n) - K(B \upharpoonright n) \geq^+ 2^{\log n - h(\log n)} - \log n \cdot 2^{h(\log n)}.$$

Thus, the set B above is g-deep, where

$$g(n) = 2^{\log n - h(\log n)} - \log n 2^{h(\log n)} > n^{1-\delta} - \log n(n^{\delta}) > n^{\epsilon},$$

for some $\epsilon < 1$. □

On the other hand, in the same paper [79], it is observed that there exists a non-empty Π_1^0 class $\mathcal{C}$ in which every member is a deep set, which (by well-known basis theorems, see [52]) implies the existence of, e.g., deep sets which are low, superlow, or of hyperimmune-free degree. As a further corollary of this result, we observe in Lemma 4.2.14 below that all DNC_2 functions (i.e., diagonally non-computable functions whose range is $\{0, 1\}$) are deep.

4.2 Relativized depth

In this part, we propose a first relativized notion of depth, in order to better understand the power of oracles in organizing information.

Definition 4.2.1. Given an oracle $A \in \mathbf{2}^{\mathbb{N}}$, we say that $X \in \mathbf{2}^{\mathbb{N}}$ is A-*deep* if, for every computable time bound t,

$$\lim_{n\to\infty} K^{A,t}(X \upharpoonright n) - K^A(X \upharpoonright n) = +\infty$$

Otherwise, we say that X is A-*shallow*.

The choice of focusing only on computable time bounds, instead of considering also A-computable ones, in the above definition is mainly due to obtain a relativized version of the Slow Growth Law that still works with tt-reductions. In other words, we want to stick to "fast" oracle computations.

For the rest of the chapter, we consider universal A-lss $\mathbf{m}^A$ (notice that Definition 4.1.3 relativizes in the obvious way). Moreover, similarly to Definition 4.1.5, given a time bound t, we let $\mathbf{m}^{A,t}$ be the function defined, for every string σ, by $\mathbf{m}^{A,t}(\sigma) = \mathbf{m}^A_{t(|\sigma|)}(\sigma)$. We notice that Lemma 4.1.6 and Theorem 4.1.7 relativize in the following way.

Lemma 4.2.2. *Let $A \in \mathbf{2}^{\mathbb{N}}$. For any given computable time bound t, $\mathbf{m}^{A,t}$ is a semimeasure and $\mathbf{m}^{A,t} \leq_{tt} A$. Conversely, if m is a semimeasure tt-below A, there exists a computable time bound t such that $m \leq^{\times} \mathbf{m}^{A,t}$. In particular, for any given computable time bound t, $2^{-K^{A,t}}$ is a semimeasure that is tt-below A, hence there exists a computable time bound t' such that $2^{-K^{A,t}} \leq^{\times} \mathbf{m}^{A,t'}$ (or equivalently, $-\log \mathbf{m}^{A,t'} \leq^{+} K^{A,t}$). Moreover, for every computable time bound t there is a computable time bound t' such that $\mathbf{m}^{A,t} \leq^{\times} 2^{-K^{A,t'}}$ (or equivalently, $K^{A,t'} \leq^{+} -\log \mathbf{m}^{A,t}$)).*

Proof. The first part of the lemma is immediate, provided one recalls that the relativized version $\mathbf{m}^A$ of the universal lower semicomputable semimeasure can (and should!) be defined uniformly, that is there is a two-place function $\Phi : \mathbf{2}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{R}^+$ such that

(i) The set $\{(A, n, q) \in \mathbf{2}^{\mathbb{N}} \times \mathbb{N} \times \mathbb{Q} \mid \Phi(A, n) > q\}$ is Σ^0_1

(ii) For every A, $n \mapsto \Phi(A, n)$ is a semi-measure

(iii) For any other function Ψ with property (i), there exists a constant $c > 0$ such that for each B, if $n \mapsto \Psi(B, n)$ is a semimeasure, then $\Phi(B, n) > c \cdot \Psi(B, n)$ for all n.

(see for example [44] for a proof of the existence of such a Φ) and thus one may *define* $\mathbf{m}^A(n)$ to be $\Phi(A, n)$ for all A and n (and $\mathbf{m}^{A,t}$ its time-bounded version, canonically defined by adding a time bound on Φ). With this definition, we do get $\mathbf{m}^{A,t} \leq_{tt} A$.

Conversely, suppose m is a semimeasure which is tt-below A. Let Ψ be the total functional such that $m = \Psi^A$. By item (iii) above, let c be a constant such that $\mathbf{m}^B(n) > c \cdot \Psi^B(n)$ for all n whenever Ψ^B is a semimeasure. This means that for all n the Π^0_1 class

$$\left\{B \mid \Psi^B \text{ is a semimeasure and } c \cdot \Psi^B(n) > \mathbf{m}^B(n)\right\}$$

must be empty, hence by effective compactness of $\mathbf{2}^{\mathbb{N}}$ one can effectively compute some $t(n)$ such that

$$\left\{B \mid \Psi^B \text{ is a semimeasure and } c \cdot \Psi^B(n) > \mathbf{m}^B(n)[t(n)]\right\} = \emptyset$$

Since $\Psi^A = m$ is a semimeasure, it follows that $c \cdot m \leq \mathbf{m}^{A,t}$.

Finally, the last statement is the relativization to A of Theorem 4.1.7. $\square$

Now, it is possible to prove that the following properties holds for relativized depth, by relativizing the proofs of the corresponding properties of unrelativized depth (namely Propositions 4.1.13, 4.1.14 and 4.1.17).

Proposition 4.2.3. *Let $A \in \mathbf{2}^{\mathbb{N}}$.*

(i) (Relativized SGL) Let $X \leq_{tt} Y$ and X be A-deep. Then Y is A-deep.

(ii) Every A-ML-random set is A-shallow.

(iii) Every A-tt-computable set is A-shallow.

(iv) A' is A-deep.

Next theorem shows how relativized depth is preserved when considering different oracles.

Theorem 4.2.4. *Let $A, B \in \mathbf{2}^{\mathbb{N}}$ such that $A \leq_{tt} B$ and $A \equiv_T B$. Then every B-deep set is also A-deep.*

Proof. Let $X \in \mathbf{2}^{\mathbb{N}}$ be A-shallow. Hence, there is a computable time bound t such that

$$\left(\overset{\infty}{\exists} n\right) \left[K^{A,t}(X \upharpoonright n) =^{+} K^{A}(X \upharpoonright n)\right].$$

Since $A \equiv_T B$, by Lemma 4.1.2 (i), we get

$$(\forall n) \left[K^{A}(X \upharpoonright n) =^{+} K^{B}(X \upharpoonright n)\right].$$

Moreover, by Lemma 4.1.2 (ii), since $A \leq_{tt} B$, we also have that

$$(\forall n)\left[K^{B,t'}(X \upharpoonright n) \leq^{+} K^{A,t}(X \upharpoonright n)\right].$$

Hence, we get

$$\left(\overset{\infty}{\exists} n\right)\left[K^{B,t'}(X \upharpoonright n) \leq^{+} K^{A,t}(X \upharpoonright n) =^{+} K^{A}(X \upharpoonright n) =^{+} K^{B}(X \upharpoonright n)\right],$$

so that X is B-shallow. $\square$

Notice that usually, in computability theory, relativizing a class $\mathcal{C}$ defines a class $\mathcal{C}^A$ such that either $\mathcal{C}^A \subseteq \mathcal{C}$ (e.g., when $\mathcal{C}$ is the class of ML-random sets) or $\mathcal{C} \subseteq \mathcal{C}^A$ (e.g., when $\mathcal{C}$ is the class of computable sets), for all oracles A.

Being defined in terms of two quantities which decrease mutually independently when an oracle is applied, this is not the case of the classes of deep and shallow sets. A priori, for an oracle A, we have four possible different scenarios:

1. The classes of A-deep sets and deep sets are incomparable, meaning that there are both shallow but A-deep sets and deep but A-shallow sets. In section 4.2.1, we show that $\emptyset'$ is an example for this scenario.

2. All deep sets remain deep relative to A, but there are shallow sets which look deep relative to A, so that depth implies A-depth, but the reverse implication fails: Section 4.2.2 is devoted to show that this is the case of ML-random oracles.

3. All shallow sets remain shallow relative to A, but there are deep sets which look shallow relative to A, hence A-depth implies depth, but the converse does not hold.

4. The class of A-deep sets and the one of deep sets coincide.

In Section 4.2.4, we observe that K-trivial oracles are examples of either scenario 3 or 4. However, while the latter obviously applies to all computable sets, we do not know which of them holds for uncomputable K-trivial oracles. Notice that, a priori, different K-trivial oracles may give different answers.

4.2.1 $\emptyset'$-depth

In this section, we build a Δ_2^0 set which is $\emptyset'$-deep but ML-random, hence shallow. This construction shows, in particular:

(i) There are oracles A such that some A-computable sets are A-deep. (Notice that, by Proposition 4.2.3 (iii), this is not the case for A-*tt*-computable sets, namely sets computable with oracle A within some computable time bound).

(ii) There are oracles A such that the class of A-deep sets is incomparable with the corresponding unrelativized class: indeed, every c.e. deep set (including $\emptyset'$) is clearly $\emptyset'$-shallow (as *tt*-below $\emptyset'$), while the set we construct in Theorem 4.2.6 is $\emptyset'$-deep but shallow.

The construction is based on the following technical lemma (stated in the following form in [66, Remark 3.1], see also [43]).

Lemma 4.2.5 (Space Lemma)**.** *For a rational $\delta > 1$ and positive integer k, let $l(\delta, k) = \lceil \frac{k+1}{1-1/\delta} \rceil$. For every martingale d and $\sigma \in \mathbf{2}^{<\mathbb{N}}$,*

$$\left|\left\{\tau \in \mathbf{2}^{l(\delta,k)} : \ d(\sigma\tau) < \delta d(\sigma)\right\}\right| \geq k.$$

Let us now prove the main theorem of this section.

Theorem 4.2.6. *There exists a set $X \in \Delta_2^0$ such that X is ML-random and X is $\emptyset'$-deep.*

Proof. Let $\mathbf{d}$ be a universal lower-semicomputable martingale and let $T : \mathbb{N} \to \mathbb{N}$ be a Δ_2^0 function such that for any computable time bound t we have $\overset{\infty}{\forall} n \ t(n) \leq T(n)$. For $n > 0$ let $\delta_n = 1 + 1/n^2$.

The construction of X will be done by a finite extension method, that is, we will build X as the limit of an increasing (with respect to the prefix relation) sequence of strings (τ_n), the construction being effective in $\emptyset'$. We set σ_0 to be the empty string λ. Having built σ_n, σ_{n+1} is chosen as follows. By the Space Lemma, the set

$$A_{n+1} = \left\{\tau \in \mathbf{2}^{l(\delta_{n+1}, 2^{n+1})} : \ \mathbf{d}(\sigma_n\tau) < \delta_{n+1} \cdot \mathbf{d}(\sigma)\right\}$$

has at least 2^{n+1} elements. Thus, there must exist some $\tau \in A_{n+1}$ such that $K^{\emptyset'}(\sigma_n\tau) > n$ and a fortiori there must exist some τ in A_{n+1} such that

$K^{\emptyset',F}(\sigma_n\tau) > n$. Since the latter is a $\emptyset'$-decidable property and A_{n+1} is $\emptyset'$-c.e, one can $\emptyset'$-effectively find such a τ and set $\sigma_{n+1} = \sigma_n\tau$.

We verify that this construction works via a series of claims.

Claim 5. *X is Δ^0_2*

Proof of claim. This is clear because the whole construction is effective relative to $\emptyset'$. ■

Claim 6. *For all n, $\mathbf{d}(\sigma_n) \leq \mathbf{d}(\lambda) \cdot \prod_{i=1}^{n}(1+1/i^2)$. Thus the sequence $\mathbf{d}(\sigma_n)$ is bounded, and thus X is Martin-Löf random.*

Proof of claim. The inequality is proved by induction. It is obvious for $n = 0$ and if we have it for some n, then by choice of τ and σ_{n+1}, we have

$$\mathbf{d}(\sigma_{n+1}) \leq \delta_{n+1}\mathbf{d}(\sigma_n) \leq (1+1/(n+1)^2)\mathbf{d}(\sigma_n) \leq (1+1/(n+1)^2)\cdot\mathbf{d}(\lambda)\cdot\prod_{i=1}^{n}(1+1/i^2)$$

which finishes the induction. Thus we have, for all n, $\mathbf{d}(\sigma_n)$ is bounded by $\mathbf{d}(\lambda) \cdot \prod_{i=1}^{\infty}(1 + 1/i^2)$ which is a finite real. Since all σ_n are initial segments of X, this means that $\liminf_k \mathbf{d}(X \upharpoonright k)$ is finite, hence X is Martin-Löf random. ■

Claim 7. *$|\sigma_n| = n^2/2 + O(n \log n)$.*

Proof of claim. Indeed we have by construction

$$|\sigma_{n+1}| = |\sigma_n| + l(\delta_{n+1}, 2^{n+1}) = |\sigma_n| + \left\lceil \log\left(\frac{2^{n+1}+1}{1-\frac{1}{1+1/(n+1)^2}}\right)\right\rceil = |\sigma_n| + n + O(\log n)$$

and thus, by summation, $|\sigma_n| = \frac{n^2}{2} + O(n \log n)$. ■

Claim 8. *For any computable time bound t, $K^{\emptyset',t}(\sigma_n) \geq n$ while $K^{\emptyset'}(\sigma_n) = O(\log n)$*

Proof of claim. Let t be a computable time bound. For almost all k, $t(k) \leq F(k)$ hence for almost all τ, $K^{\emptyset',t}(\tau) \geq K^{\emptyset',F}(\tau)$. In particular, for almost all n, $K^{\emptyset',t}(\sigma_n) \geq K^{\emptyset',F}(\sigma_n) \geq n$ (the last inequality is by choice of σ_{n+1} in the construction). On the other hand, since the sequence of σ_n is $\emptyset'$-computable, we have $K^{\emptyset'}(\sigma_n) =^+ K^{\emptyset'}(n) = O(\log n)$. ■

Claim 9. *X is $\emptyset'$-deep*

Proof of claim. Let t be a computable time bound. For any k, by construction of X, there is an n such that

$$\sigma_n \preceq (X \upharpoonright k) \preceq \sigma_{n+1}$$

We can recover σ_n from $|\sigma_n|$ and $X \upharpoonright k$ by just truncating the latter to its first $|\sigma_n|$ bits. Since $|\sigma_n|$ is computable in n, this means that if t' is sufficiently fast-growing,

$$K^{\emptyset',t}(X \upharpoonright k) \geq K^{\emptyset',t'}(\sigma_n) - K^{\emptyset',t}(n) \geq n - O(\log n)$$

(the last inequality following from Claim 4).

By the same reasoning, we can recover $X \upharpoonright k$ from σ_{n+1} and k, hence

$$K^{\emptyset'}(X \upharpoonright k) \leq K^{\emptyset'}(\sigma_{n+1}) + K^{\emptyset'}(k) \leq O(\log n) + O(\log k)$$

Moreover, since $|\sigma_n| \leq k \leq |\sigma_{n+1}|$, by Claim 3, $k \sim n^2/2$, meaning that one can replace $O(\log k)$ by $O(\log n)$ in the above expression. Putting both inequalities together, we get

$$K^{\emptyset',t}(X \upharpoonright k) - K^{\emptyset'}(X \upharpoonright k) \geq n - O(\log n) \geq \sqrt{2k} - o(\sqrt{k})$$

which finally shows that

$$K^{\emptyset',t}(X \upharpoonright k) - K^{\emptyset'}(X \upharpoonright k) \to \infty$$

■

This finishes the proof. □

4.2.2 Depth relative to ML-random oracles

The main goal of this section is to prove that depth is strictly implied by depth relative to any ML-random oracle. We will first prove that ML-random sets are non-trivial examples of oracles for which all deep sets remain deep relative to them. We successively show that shallowness is a notion preserved by almost every oracle, namely that if a set is shallow, then it is shallow relatively to a class of oracles of measure 1. Despite this fact, every ML-random set "adds" deep sets: for every ML-random oracle we can find a shallow set which is deep relatively to that oracle. Interestingly, as a consequence of the existence of such sets, we can give a quite short and easy proof of the fact, originally proved by Barmpalias, Lewis and Ng [7], that every PA-complete degree is the join of two ML-random degrees.

Deep sets remain deep relative to ML-random oracles

In order to prove that no deep set can be shallow relative to ML-random oracles, we recall another characterization of ML-randomness.

Definition 4.2.7. $\Psi : \mathbf{2}^{\mathbb{N}} \to [0, \infty]$ is an integral test if

- Ψ is *lower-semicomputable*, i.e. it is the supremum of a computable sequence of computable functions $\Psi_n : \mathbf{2}^{\mathbb{N}} \to [0, \infty)$, and
- $\int_{X \in \mathbf{2}^{\mathbb{N}}} \Psi(X)\, d\mu \leq 1$.

Integral tests characterize ML-randomness, in the following sense. (For a proof of the statement below, see [15]).

Proposition 4.2.8. *X is not ML-random if and only if there is an integral test Ψ such that $\Psi(X) = \infty$.*

We can now turn to the main result.

Theorem 4.2.9. *Let $A \in \mathbf{2}^{\mathbb{N}}$ be ML-random. Then every deep set is A-deep.*

Proof. We prove, in particular, that, if $A \in \mathbf{2}^{\mathbb{N}}$ is ML-random, then for every computable time bound t there exists a computable time bound t' such that

$$(\forall \sigma) \left[K^{t'}(\sigma) - K(\sigma) \leq^{+} K^{A,t}(\sigma) - K^{A}(\sigma) \right]$$

which will immediately imply the theorem.

For any string σ and time bound t, let $\mathbf{m}(\sigma) = 2^{-K(\sigma)}$, $\mathbf{m}^t(\sigma) = 2^{-K^t(\sigma)}$, $\mathbf{m}^A(\sigma) = 2^{-K^A(\sigma)}$ and $\mathbf{m}^{A,t}(\sigma) = 2^{-K^{A,t}(\sigma)}$.

For every $A \in \mathbf{2}^{\mathbb{N}}$ and every computable time bound t, it clearly holds that

$$\sum_{\sigma \in \mathbf{2}^{<\mathbb{N}}} \mathbf{m}^{A,t}(\sigma) \leq \sum_{\sigma \in \mathbf{2}^{<\mathbb{N}}} \mathbf{m}^{A}(\sigma) \leq 1.$$

Hence,

$$\int_{A \in \mathbf{2}^{\mathbb{N}}} \sum_{\sigma \in \mathbf{2}^{<\mathbb{N}}} \mathbf{m}^{A,t}(\sigma)\, d\mu \leq 1$$

and, by interchanging the sum with the integral,

$$\sum_{\sigma \in \mathbf{2}^{<\mathbb{N}}} \left(\int_{A \in \mathbf{2}^{\mathbb{N}}} \mathbf{m}^{A,t}(\sigma)\, d\mu \right) \leq 1,$$

which means that the map $\sigma \mapsto \int_{A\in\mathbf{2}^{\mathbb{N}}} \mathbf{m}^{A,t}\, d\mu$ is a discrete semimeasure. Moreover, it is computable. Thus, by Lemma 4.1.6,

$$\int_{A\in\mathbf{2}^{\mathbb{N}}} \mathbf{m}^{A,t}(\sigma)\, d\mu \leq c \cdot \mathbf{m}^{t'}(\sigma),$$

for some computable time bound t'.

Now, consider the map $\Psi : \mathbf{2}^{\mathbb{N}} \to [0,\infty]$ given by

$$\Psi(A) = \sum_{\sigma\in\mathbf{2}^{<\mathbb{N}}} \frac{\mathbf{m}^{A,t}(\sigma)\mathbf{m}(\sigma)}{c \cdot \mathbf{m}^{t'}(\sigma)}.$$

Then Ψ is lower-semicomputable and, by the above computations,

$$\int_{A\in\mathbf{2}^{\mathbb{N}}} \Psi(A)\, d\mu \leq 1,$$

that is, Ψ is an integral test. Let A be a ML-random set, so that $\Psi(A) < k$ for some $k \in \mathbb{N}$. By definition of Ψ this means that

$$\sum_{\sigma\in\mathbf{2}^{<\mathbb{N}}} \frac{\mathbf{m}^{A,t}(\sigma)\mathbf{m}(\sigma)}{c \cdot \mathbf{m}^{t'}(\sigma)} < k$$

and thus $\sigma \mapsto \frac{\mathbf{m}^{A,t}(\sigma)\mathbf{m}(\sigma)}{kc\cdot\mathbf{m}^{t'}(\sigma)}$ is an A-lss. It follows that for every $\sigma \in \mathbf{2}^{<\mathbb{N}}$,

$$\frac{\mathbf{m}^{A,t}(\sigma)\mathbf{m}(\sigma)}{\mathbf{m}^{t'}(\sigma)} \leq^{\times} \mathbf{m}^{A}(\sigma),$$

which, by taking the logarithm of both sides, implies

$$K^{t'}(\sigma) - K(\sigma) \leq^{+} K^{A,t}(\sigma) - K^{A}(\sigma),$$

which is what we wanted. □

As a consequence, we get our first example of an oracle making the class of deep sets strictly smaller.

Corollary 4.2.10. *The class of Ω-deep sets strictly contains the class of deep sets.*

Proof. Since Ω is ML-random, every deep set is Ω-deep. It is then enough to find a set X which is shallow but Ω-deep. Let X be the set built in Theorem 4.2.6. Recall that $\Omega \leq_{tt} \emptyset'$ and $\Omega \equiv_T \emptyset'$, hence, by Theorem 4.2.4, X is Ω-deep. □

A shallow sets which is deep relative to a ML-random oracle

This section is mainly devoted to showing that what we observe for Ω in Corollary 4.2.10 holds, in fact, for every ML-random oracle. To show this fact, we need to recall some terminology.

Definition 4.2.11. $f : \mathbb{N} \to \mathbb{N}$ is a *Solovay function* if

$$(\forall n)[f(n) \leq^{+} K(n)] \quad \text{and} \quad \left(\overset{\infty}{\exists} n\right)[K(n) \leq^{+} f(n)].$$

As shown in [48], for any superlinear time bound t, the time-bounded Kolmogorov complexity K^t is a computable Solovay function.

We first observe that any shallow set remains shallow relative to almost every oracle, where "almost every" is meant in the measure-theoretic sense. Namely, we have the following result.

Theorem 4.2.12. *If $X \in \mathbf{2}^{\mathbb{N}}$ is shallow, then*

$$\mu\left(\{A : \ X \textit{ is } A\textit{-shallow}\}\right) = 1.$$

In particular, X is A-shallow for every A which is 2-random relative to X (i.e. $A \notin \cap_n \mathcal{U}_n^X$, where $(\mathcal{U}_n^X)_{n \in \mathbb{N}}$ is any sequence of uniformly Σ_2^X classes, with $\mu(\mathcal{U}_n^X) \leq 2^{-n}$ for all n).

Proof. Since X is shallow, there are a computable time bound t and $c \in \mathbb{N}$ such that

$$\left(\overset{\infty}{\exists} n\right)\left[\frac{\mathbf{m}(X \upharpoonright n)}{\mathbf{m}^t(X \upharpoonright n)} < c\right].$$

Recall that the map

$$\sigma \mapsto \int_{A \in \mathbf{2}^{\mathbb{N}}} \mathbf{m}^A(\sigma)\, d\mu$$

is a lower-semicomputable discrete semimeasure. Hence, by the universality of $\mathbf{m}$, we get that $\mathbf{m}(\sigma) \geq^{\times} \int_{A \in \mathbf{2}^{\mathbb{N}}} \mathbf{m}^A(\sigma)\, d\mu$ for all $\sigma \in \mathbf{2}^{<\mathbb{N}}$. On the other hand, there exists a constant c' such that $c'\mathbf{m}^A > \mathbf{m}$ for any A, hence $\int_{\mathbf{2}^{\mathbb{N}}} \mathbf{m}^A(\sigma) \geq^{\times} \mathbf{m}(\sigma)$. Putting the two together,

$$\mathbf{m}(\sigma) =^{\times} \int_{A \in \mathbf{2}^{\mathbb{N}}} \mathbf{m}^A(\sigma)\, d\mu. \tag{4.3}$$

For any $k \in \mathbb{N}$ let

$$\mathcal{L}_k = \left\{ Y : \left(\overset{\infty}{\forall} n\right) \left[\frac{\mathbf{m}^Y(X \upharpoonright n)}{\mathbf{m}^t(X \upharpoonright n)} \geq k \right] \right\}.$$

Now, suppose that X is A-deep, so that, in particular,

$$\left(\forall k \overset{\infty}{\forall} n\right) \left[\frac{\mathbf{m}^A(X \upharpoonright n)}{\mathbf{m}^t(X \upharpoonright n)} \geq k \right],$$

meaning that $A \in \mathcal{L}_k$ for every k. It is then enough to show that

$$\mu \left(\bigcap_{k \in \mathbb{N}} \mathcal{L}_k \right) = 0,$$

namely that

$$\lim_{k \to \infty} \mu(\mathcal{L}_k) = 0.$$

Observe that

$$\mathcal{L}_k = \liminf_{n \to \infty} \mathcal{U}_n^k,$$

where

$$\mathcal{U}_n^k = \left\{ Y : \left[\frac{\mathbf{m}^Y(X \upharpoonright n)}{\mathbf{m}^t(X \upharpoonright n)} \geq k \right] \right\}.$$

We claim that

$$\left(\overset{\infty}{\exists} n\right) \left[\mu\left(\mathcal{U}_n^k\right) \leq^{\times} \frac{1}{k} \right].$$

Indeed, since K^t is a Solovay function,

$$\left(\overset{\infty}{\exists} n\right) \left[\mathbf{m}^t(X \upharpoonright n) \geq^{\times} \mathbf{m}(X \upharpoonright n) \right].$$

Then, for any such n, if $Y \in \mathcal{U}_n^k$, then

$$\mathbf{m}^Y(X \upharpoonright n) \geq^{\times} k \cdot \mathbf{m}^t(X \upharpoonright n) \geq^{\times} k \cdot \mathbf{m}(X \upharpoonright n) \geq^{\times} k \cdot \int_{A \in \mathbf{2}^{\mathbb{N}}} \mathbf{m}^A(X \upharpoonright n)\, d\mu,$$

where the last inequality follows from (4.3). Our claim follows then by Markov's inequality. Therefore, we get

$$\mu(\mathcal{L}_k) = \mu(\liminf_{n \to \infty} \mathcal{U}_n^k) \leq \liminf_{n \to \infty} \mu(\mathcal{U}_n^k) \leq^{\times} \frac{1}{k},$$

where the first inequality follows by Fatou's Lemma, and hence

$$\lim_{k\to\infty} \mu(\mathcal{L}_k) \leq^{\times} \lim_{k\to\infty} \frac{1}{k} = 0.$$

Finally, observe that $\mathcal{L}_k$ is the limit inferior of a uniformly c.e. sequence of $\Sigma_1^{0,X}$ classes with bounded measure. Hence, as proven in [9], each $\mathcal{L}_k$ is contained in a $\Sigma_2^{0,X}$ class of bounded measure. Hence, the classes $\mathcal{L}_k$ form a test for 2-randomness relative to X, so that X must be A-shallow for any A which is 2-random relative to X. $\square$

It is then natural to ask whether for a "very random" oracle every set must be shallow. In other words, whether there is a randomness notion sufficiently strong to make any such oracle totally useless in organizing information. We answer this question in the negative. Indeed, we show that every ML-random oracle makes some shallow set deep. Intuitively, the proof of this fact is similar to the one-time pad protocol in cryptography: we can "mix" together some important piece of information x with some random string a we know, so that the output $x \triangle a$ still looks important for us (as we can distinguish the added random noise a), while looking random to the others.

In order to prove formally our claim, we recall some well-known notions and facts.

Definition 4.2.13. Let (ϕ_e) be an effective enumeration of all partial computable functions from $\mathbb{N}$ to $\mathbb{N}$. A total function f is *diagonally non-computable* (DNC) if $f(e) \neq \phi_e(e)$ whenever $\phi_e(e)$ is defined. We say that f is DNC_2 if it is DNC and its range is $\{0, 1\}$. The set DNC_2 of such functions is a Π_1^0 class. A set X is said to be*PA-complete* if it computes some member of DNC_2.

(The terminology 'PA-complete' comes from the equivalent definition where one replaces the class DNC_2 by the class of complete coherent extensions of Peano Arithmetic).

A well-known property of the class DNC_2 is that it is *universal* (a.k.a. *Medvedev-complete*) among Π_1^0 classes, that is, for every non-empty Π_1^0 class $\mathcal{C}$, there exists a total functional Φ such that $\Phi^X \in \mathcal{C}$ for any $X \in \mathrm{DNC}_2$. Combined with a result of Moser and Stephan, this yields the following lemma.

Lemma 4.2.14. *Every $X \in DNC_2$ is deep.*

Proof. Indeed, in [79], it is shown that there exists a non-empty Π_1^0 class $\mathcal{C}$ in which every member is a deep set. Since DNC_2 is universal, every

$X \in \mathrm{DNC}_2$ tt-computes some member of $\mathcal{C}$ (via the same functional). Since depth is tt-closed upwards, this proves our result. □

Thus we can use well-known basis theorems to obtain deep sets with some desired properties. In particular, we will use the following fact (for a proof, see [38]).

Proposition 4.2.15 (Randomness Basis Theorem)**.** *Let R be a ML-random set. Every non-empty Π^0_1 class contains an element X such that R is X-ML-random.*

We are now ready to prove the main result of this section.

Theorem 4.2.16. *If a set A is ML-random, then there exists a shallow set which is A-deep.*

Proof. Let A be a ML-random set. By Proposition 4.2.15, there exists a deep set X such that A is ML-random relative to X. On the other hand, by Theorem 4.2.9, X is also A-deep. We then consider the set

$$A \triangle X = (A \setminus X) \cup (X \setminus A).$$

We first notice that, for every pair of sets S, T, clearly $K^S(S \triangle T \upharpoonright n) =^+ K^S(T \upharpoonright n)$, and the same holds when considering time bounds.

Hence, since A is ML-random relative to X, $A \triangle X$ is also ML-random relative to X, and hence shallow.

On the other hand, since X is A-deep, $(A \triangle X)$ must also be A-deep. □

4.2.3 A digression on a result about PA-complete degrees

In this section we observe that the existence of sets as in the proof of Theorem 4.2.16 can improve upon (and give a simpler proof of) a theorem of Barmpalias, Lewis and Ng ([7]), who proved that for every PA-complete A, there exist two Martin-Löf random X, Y such that $A \equiv_T X \oplus Y$. We will prove the following.

Theorem 4.2.17. *Let $A \in DNC_2$. Then there exist two Martin-Löf random X, Y such that $A = X \triangle Y$ (in particular $A \equiv_{tt} X \oplus Y$).*

In order to prove Theorem 4.2.17, we need the following property of universal Π^0_1 classes.

Lemma 4.2.18. *Let $\mathcal{C}$ be a non-empty universal Π^0_1 class.*

(i) For any $A \in DNC_2$, there is a set $B \in \mathcal{C}$ such that $A \equiv_{tt} B$

(ii) For any A which Turing-computes some DNC_2 function, there exists some $B \in \mathcal{C}$ such that $A \equiv_T B$.

Proof. (i) Let Φ witness the universality of $\mathcal{C}$, that is, $\Phi^X \in \mathrm{DNC}_2$ whenever $X \in \mathcal{C}$.

The construction of B is done via an A-effective forcing argument: we define a sequence $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \dots$ of Π^0_1 classes such that the sequence of indices for the $\mathcal{C}_i$ is A-computable, each of these classes is non-empty and that their intersection $\cap_s \mathcal{C}_s$ is a singleton which will be our B.

To ensure that $\cap_s \mathcal{C}_s$ will be singleton, we will have at each step a string σ_s such that $|\sigma_s| = s$ and $\mathcal{C}_s \subseteq [\sigma_s]$.

Let $\mathcal{C}_0 = \mathcal{C}$ and $\sigma_0 = \lambda$. Inductively, assume that we have already built non-empty Π^0_1 classes $\mathcal{C}_0 \supseteq \cdots \supseteq \mathcal{C}_s$ and strings $\sigma_0 \preceq \sigma_1 \preceq \ldots \preceq \sigma_s$ such that $\mathcal{C}_i \subseteq [\sigma_i]$ for all $i \leq s$.

Using an index for $\mathcal{C}_s$, one can compute some n_s such that $\phi_{n_s}(n_s)$ returns the first $i \in \{0,1\}$ found such that if $\mathcal{C}_s \cap [\sigma_s i] = \emptyset$ (and stays undefined if no such i is found). Since $A \in \mathrm{DNC}_2$, we have $\phi_{n_s}(n_s) \neq A(n_s)$. Let thus $\sigma_{s+1} = \sigma_s A(n_s)$, so that $\mathcal{C}_s \cap [\sigma_{s+1}]$ is non-empty.

Now, by the Recursion Theorem, let m_s be an index such that ϕ_{m_s} waits for some $i \in \{0,1\}$ to be such that

$$(\forall X \in \mathcal{C}_s \cap [\sigma_{s+1}]) \ \Phi^X(m_s) = i$$

(since $\mathcal{C}_s \cap [\sigma_{s+1}]$ is a Π^0_1 class, this property is a c.e. event) and if such an i is found returns $\phi_{m_s}(m_s) = 1 - i$ (staying undefined otherwise).

Since the image of $\mathcal{C}_s \cap [\sigma_{s+1}]$ is contained in DNC_2, one must have $\Phi^X(m_s) \neq \phi_{m_s}(m_s)$ for any $X \in \mathcal{C}_s$. This means that in fact $\phi_{m_s}(m_s)$ must be undefined, which in turns implies that for all $i \in \{0,1\}$,

$$\mathcal{C}_s \cap [\sigma_{s+1}] \cap \{X \, : \Phi^X(m_s) = i\} \neq \emptyset.$$

Now define

$$\mathcal{C}_{s+1} = \mathcal{C}_s \cap [\sigma_{s+1}] \cap \{X \, : \Phi^X(m_s) = A(s)\}$$

Finally, define B to be the unique element of $\bigcap_s \mathcal{C}_s$ (or equivalently the limit of the σ_s). We claim that B is as wanted.

To see that A computes B, observe that the whole construction is A-tt-effective. If we have already A-tt-computed an index for $\mathcal{C}_s$ and σ_s, n_s can be effectively tt-computed and $\sigma_{s+1} = \sigma_s A(n_s)$ can be computed from A. Likewise, since the Recursion Theorem is effective, m_s can then be computed, and an index for $\mathcal{C}_{s+1}$ tt-effectively obtained from m_s and A.

But the whole construction (indices for the $\mathcal{C}_s$ and strings σ_s) can also be tt-recovered from B. Indeed, σ_s is the prefix of B of length s, thus $A(n_s)$ can be tt-computed from B for all s, and to compute an index for $\mathcal{C}_{s+1}$ knowing one for $\mathcal{C}_s$ we only need to know m_s (which can be tt-computed from the index of $\mathcal{C}_s$ and σ_{s+1}) and $A(s)$ which is just $\Phi^B(m_s)$ (and recall that Φ is a tt-functional). This implies that B tt-computes the sequence of m_s and thus $A(s)$ for each s since $A(s) = \Phi^B(m_s)$.

(*ii*) The proof of this part is almost identical. If $A \geq_T F$ for some DNC_2 function, make the same construction only replacing $A(n_s)$ by $F(n_s)$. Then A computes B (not necessarily in a tt-way since it needs to compute $F(n_s)$ for all s to do so) and B tt-computes A. □

We can now prove Theorem 4.2.17.

Proof of Theorem 4.2.17. For any $k \in \mathbb{N}$, let

$$\mathrm{MLR}_k = \{X : \ (\forall n) K(X \upharpoonright n) \geq n - k\}.$$

(which is a Π^0_1 class). By the characterization of Martin-Löf randomness in terms of prefix-free Kolmogorov complexity, a set X is Martin-Löf random if and only if $X \in \mathrm{MLR}_k$ for some large enough k.

In the proof of Theorem 4.2.16 we saw that for k large enough, the Π^0_1 class

$$\mathcal{C} = \{F \oplus X \oplus Y \ : F \in \mathrm{DNC}_2, \ X, Y \in \mathrm{MLR}_k, \ F = X \triangle Y\}$$

is non-empty. Moreover, the class $\mathcal{C}$ is universal , as witnessed by the first projection. Let $A \in \mathrm{DNC}_2$. By the previous lemma, $A \equiv_{tt} F \oplus X \oplus Y$ for some $F \oplus X \oplus Y \in \mathcal{C}$. Clearly, $X \oplus Y \leq_{tt} F \oplus X \oplus Y$. On the other hand, since $F = X \triangle Y$, we also have that $F \leq_{tt} X \oplus Y$ and hence $A \equiv_{tt} F \oplus X \oplus Y \equiv_{tt} X \oplus Y$.

□

Corollary 4.2.19 (Barmpalias, Lewis, Ng)**.** *If A has PA-complete Turing degree, there are two Martin-Löf random sets X, Y such that $A \equiv_T X \oplus Y$.*

Proof. Since A has PA-complete Turing degree, by Lemma 4.2.18 (where $\mathcal{C}$ is taken to be the class DNC_2), A is Turing equivalent to some $\tilde{A} \in \mathrm{DNC}_2$. By the previous theorem, $\tilde{A}$ is in turn tt-equivalent (hence Turing equivalent) to the join of two Martin-Löf random sets. $\square$

4.2.4 An open question about K-trivial oracles

The intuitive notion of *being far from being ML-random* turned out to be a central notion in algorithmic randomness. This is formally expressed by the property of K-triviality, which we have recalled in Definition 4.1.15 above. Intuitively speaking, describing a prefix of a K-trivial set is at most as hard as describing its length. Another lowness property related to prefix-free complexity is when an oracle does not help in further compressing strings. Moreover, as already mentioned, Moser and Stephan proved in [79] that every K-trivial set is shallow.

It is easy to realize that every shallow set remains shallow relatively to any K-trivial oracle.

Theorem 4.2.20. *Let $A \in \mathbf{2}^{\mathbb{N}}$ be K-trivial. Then every shallow set is A-shallow.*

Proof. Let $X \in \mathbf{2}^{\mathbb{N}}$ be shallow. Hence, there is a computable time bound t such that

$$\left(\overset{\infty}{\exists} n\right) \left[K^t(X \upharpoonright n) =^+ K(X \upharpoonright n)\right].$$

Then, for any such n,

$$K^{A,t}(X \upharpoonright n) \leq^+ K^t(X \upharpoonright n) =^+ K(X \upharpoonright n) \leq^+ K^A(X \upharpoonright n),$$

where the last inequality follows because every K-trivial set is low for K (Proposition 4.1.16). Hence, X is A-shallow. $\square$

It follows from the previous result that the class of deep sets relative to a K-trivial oracle is never larger than the class of deep sets. The following natural question remains open.

Question 4.2.21. Let A be a K-trivial set. Does A-depth always strictly imply depth? Or does A-depth and depth always coincide? Or does the answer depend on the particular oracle?

4.3 Turing-relativized depth

In this last part of the chapter, we investigate a further notion of relativized depth, in which we consider not only computable time bounds, but more generally time bounds which are Turing computable in the given oracle, thus the choice of the name *Turing-relativized-depth.* Formally, we give the following definition, which is, in some sense, the "full" relativization of Definition 4.1.11 to an oracle A.

Definition 4.3.1. Given an oracle $A \in \mathbf{2}^{\mathbb{N}}$, we say that $X \in \mathbf{2}^{\mathbb{N}}$ is *A-T-deep* if, for every time bound $t \leq_T A$,

$$\lim_{n\to\infty} K^{A,t}(X \upharpoonright n) - K^{A}(X \upharpoonright n) = +\infty$$

Otherwise, we say that X is *A-T-shallow.*

Note that, a priori, the class of A-deep sets, as defined in Definition 4.2.1, contains the class of A-T-deep sets: that is, for every oracle A, every A-T-deep set is, in particular, A-deep.

By a straightforward relativization of Lemma 4.1.6 and Theorem 4.1.7, we get the following result.

Lemma 4.3.2. *Let $A \in \mathbf{2}^{\mathbb{N}}$. For any given time bound $t \leq_T A$, $\mathbf{m}^{A,t}$ is a semimeasure and $\mathbf{m}^{A,t} \leq_T A$. Conversely, if $m \leq_T A$ is a discrete semimeasure, there exists a computable time bound t such that $m \leq^{\times} \mathbf{m}^{A,t}$. In particular, for any time bound $t \leq_T A$, $2^{-K^{A,t}}$ is an A-computable semimeasure, hence there exists a time bound $t' \leq_T A$ such that $2^{-K^{A,t}} \leq^{\times} \mathbf{m}^{A,t'}$ (or equivalently, $-\log \mathbf{m}^{A,t'} \leq^{+} K^{A,t}$). Moreover, for any time bound $t \leq_T A$, there exists a time bound $t' \leq_T A$ such that $\mathbf{m}^{A,t} \leq^{\times} 2^{-K^{A,t'}}$ (or, equivalently $K^{A,t'} \leq^{+} -\log \mathbf{m}^{A,t}$).*

As an immediate consequence, we have that $X \in \mathbf{2}^{\mathbb{N}}$ is A-T-deep if and only if, for every time bound $t \leq_T A$,

$$\lim_{n\to\infty} \frac{\mathbf{m}^{A}(X \upharpoonright n)}{\mathbf{m}^{A,t}(X \upharpoonright n)} = +\infty,$$

or also, if and only if, for every A-computable semimeasure m,

$$\lim_{n\to\infty} \frac{\mathbf{m}^{A}(X \upharpoonright n)}{m(X \upharpoonright n)} = +\infty.$$

Finally, a straightforward relativization of Theorem 4.1.12 shows that X is A-T-deep if and only if, for every time bound $t \leq_T A$ and constant c,

$$\left(\overset{\infty}{\forall} n \forall \tau \in \mathcal{U}^A[t]^{-1}(X \upharpoonright n)\right) \left[K^A(\tau) \leq |\tau| - c\right].$$

Recall that a set A is called *computably dominated* if, for each function $g \leq_T A$, there is a computable function f which dominates g, namely

$$\left(\overset{\infty}{\forall} n\right) [g(n) \leq f(n)].$$

In the literature such sets are also called of *hyperimmune-free degree*, because of the fact that a set A is not computably dominated if and only if there exists an hyperimmune set $X \equiv_T A$ ([67]). It is easy to see that, if A is a computably dominated set and $f \leq_T A$, then $f \leq_{tt} A$ (see, e.g., [75, Proposition 1.5.11]). It is also immediate to check that, if A is computably dominated, then the classes of A-deep sets and of A-T-deep sets coincide.

It is a trivial matter to check that, similarly to A-depth, A-T-depth is also preserved upwards under tt-reductions. However, it is natural to ask whether there is a strictly larger class of reductions for which the Slow Growth Law for A-T-depth applies. In this section, we positively answer this question. We start by looking at the following class of reductions, which can be regarded as tt-functionals with respect to the oracle A.

Definition 4.3.3. Given an oracle $A \in \mathbf{2}^{\mathbb{N}}$ we say that a set X is $tt(A)$-reducible to a set Y (and write $X \leq_{tt(A)} Y$), if there is an oracle Turing machine φ and a time bound $t \leq_T A$ such that $\varphi^{Y \oplus A}(n)[t(n)] \downarrow = X(n)$.

The following lemma shows a characterization of $tt(A)$-reductions similar to the characterization of tt-reductions in terms of total oracle machines.

Lemma 4.3.4. *$X \leq_{tt(A)} Y$ if and only if there is an oracle machine Φ which is total on every oracle of the form $S \oplus A$ and such that $\Phi^{Y \oplus A} = X$.*

Proof. Assume that $X \leq_{tt(A)} Y$ via the oracle machine φ and the time bound $t \leq_T A$. Then, for every $S \in \mathbf{2}^{\mathbb{N}}$ and n, it suffices to define $\Phi^{S \oplus A}(n) = \varphi^{S \oplus A}(n)$, if $\varphi^{S \oplus A}(n)$ halts within $t(n)$ steps, or $\Phi^{S \oplus A}(n) = 0$, otherwise.

Conversely, assume that the oracle machine Φ is total on every oracle of the form $S \oplus A$ and that $\Phi^{Y \oplus A} = X$. For every n, consider the set

$$T_n = \left\{\sigma \prec S \oplus A : \; S \in \mathbf{2}^{\mathbb{N}} \;\&\; \Phi^{\sigma}(n) \downarrow\right\} :$$

by compactness, for every n we can A-compute $t(n)$ such that, for every $\sigma \in T_n$, $\Phi^\sigma(n)$ halts in at most $t(n)$ steps. In particular, $\Phi^{Y \oplus A}(n)[t(n)] = X(n)$ for all n. $\square$

Trivially, $X \leq_{tt} Y$ implies $X \leq_{tt(A)} Y$. On the other hand, for every hyperimmune set A, $A \leq_{tt(A)} \emptyset$, but clearly $A \not\leq_{tt} \emptyset$.

Straightforward relativizations of the proofs of Propositions 4.1.13, 4.1.14 and 4.1.17 gives the following results.

Proposition 4.3.5. *Let $A \in \mathbf{2}^{\mathbb{N}}$.*

(i) (Slow Growth Law for A-T-depth) Let X be A-T-deep and $X \leq_{tt(A)} Y$. Then Y is also A-T-deep.

(ii) Every A-ML-random set is A-T-shallow.

(iii) Every A-computable set is A-T-shallow.

(iv) A' is A-T-deep.

We conclude by showing that, whenever A is high, the class of A-T-deep sets is strictly contained in the one of A-deep sets. Indeed, in every high degree we can construct an A-computable (hence, A-T-shallow) but A-deep set. Notice that this construction is similar in spirit to the one of Theorem 4.1.18 above, as it is based on computing a dominating function T and diagonalizing against all too long T-fast codes.

Theorem 4.3.6. *Let $A \in \mathbf{2}^{\mathbb{N}}$ be high. Then there exist a set $X \equiv_T A$ (in particular, X is A-T-shallow) which is A-deep.*

Proof. Since A is high, there is a function $T \leq_T A$ such that T dominates every computable time bound. Partition $\mathbb{N}$ into consecutive intervals $I_0 = \{0,1\}, I_1, \dots$ so that each interval I_m contains exactly 2^{m-1} elements, for $m > 0$. Moreover, for any m, define

$$P_m = \left\{\tau : |\tau| < |I_m| - 1 \;\&\; \mathcal{U}^A(\tau)[T(\max I_{m+1})] \downarrow= \sigma \text{ with } |\sigma| \in I_{m+1}\right\}.$$

We define a set X by specifying its bits at each interval I_m, which are denoted by $X[I_m]$, as follows. The first bit of $X[I_m]$ is used to code A, so that $A \leq_T X$. The remaining $|I_m|-1$ bits are used to diagonalize against the at most $2^{|I_m|-1}$ codes in P_m: in fact, if $\mathcal{U}^A(\tau)[T(\max I_{m+1}))] \upharpoonright \max I_{m+1} \neq$

$X \upharpoonright \max I_{m+1}$ for all $\tau \in P_m$, then $K^{A,T}(X \upharpoonright n) > |I_m| - 1 = 2^m - 1$, for all $n \in I_{m+1}$. Hence, for each $i \in I_m$, we compute

$$P_i = \{\tau \in P_m : \mathcal{U}^A(\tau)(j) = X(j), \text{ for all } j \in I_m \text{ with } j < i\},$$

and let $B(i) = 1$ if and only if $\mathcal{U}^A(\tau)(i) = 0$ for at least half of the codes $\tau \in P_i$.

As T dominates all computable time bounds we get that, for every computable time bound t and sufficiently large m,

$$K^{A,t}(X \upharpoonright n) > K^{A,T}(X \upharpoonright n) > |I_m| - 1 = 2^{m-1} - 1,$$

for all $n \in I_{m+1}$. Notice that, if $n \in I_{m+1}$, then $2^m \leq n \leq 2^{m+1}$, therefore we have $\lceil \log n \rceil = m + 1$ and, for simplicity, we set $\log n = \lceil \log n \rceil - 1 = m$. On the other hand, since $X \leq_T A$, we have $K^A(X \upharpoonright n) \leq^+ 2 \log n$. Thus, at the end we get that, for every computable time bound t and almost all n,

$$K^{A,t}(X \upharpoonright n) - K^A(X \upharpoonright n) > 2^{\log n} - 2 \log n \geq^+ n - 2 \log n,$$

which is eventually larger than any constant. Hence, X is A-deep. $\square$

Chapter 5

On the comparison between deterministic and probabilistic forecasting schemes

Is it possible for a gambler using a probabilistic betting strategy to become arbitrarily rich when all gamblers betting according to a deterministic strategy earn only a bounded capital?

We investigate this question in the context of algorithmic randomness, introducing the new notion of *almost everywhere computable randomness*.

The theory of algorithmic randomness aims at formalizing the intuitive concept of randomness for single outcomes, namely for infinite binary sequences. Obviously, from the perspective of classical computability theory, any infinite binary sequence drawn at random (with respect to the uniform distribution) has the same probability 0 to occur. Yet, as already mentioned in the previous chapter, there are many possible approaches to formalize effective notions of randomness, namely different ways to formalize the intuitive idea that effectively random outcomes are those which do not posses any untypical property which can be effectively tested. A popular way to do so is to look for "unpredictable" sequences: roughly speaking, we may consider random any sequence whose bits cannot be predicted with better-then-average accuracy. More precisely, we fix a certain class $\mathcal{C}$ of effective gambling strategies, which are usually called *martingales* in this context, for the following game. The bits of an infinite sequence X are revealed in ascending order. When the martingale $d \in \mathcal{C}$ has already seen n many bits of X, d bets a certain amount α of its capital that the $n+1$-th bit of X is, say, 0: if d is right, then d wins α, otherwise d loses α. We say that the martingale d *succeeds* on X if its capital tends to infinity throughout the infinite game above, and we consider a sequence X *random* (with respect to the given class $\mathcal{C}$) if no martingale in $\mathcal{C}$ succeeds on X. In particular, we talk of *computable randomness* when we consider only total computable martingales, and of *partial computable randomness* if we also allow partial computable ones. In both cases, however, these strategies are deterministic.

In our framework, instead, we also consider probabilistically effective betting strategies: intuitively speaking, we consider effective betting strategies which, in addition, are allowed to flip a fair coin before placing their bet (and possibly betting accordingly). More formally, we assume that the infinite sequence Y of coin tosses has been drawn in advance and given as an oracle to a partial computable martingale d (thus obtaining a Y-computable martingale which we denote by d^Y): hence, we say that a sequence X is *almost everywhere computably random* if, for any partial oracle martingale d, we have that

$$\mu\left(\left\{Y : \ d^Y \text{ is total and succeeds on } X\right\}\right) = 0.$$

We show that probabilistic martingales are in fact stronger than deterministic ones, by building a partial computable random sequence which is not almost everywhere computably random. It is worth noticing that this is an unusual and unexpected result in computability theory, because of a classical theorem stating that every sequence which can be computed by a probabilistic algorithm with positive probability is in fact deterministically computable ([31]). We also prove the separation between a.e. computable randomness and partial computable randomness, which happens exactly in the uniformly almost everywhere dominating Turing degrees.

We should also note that probabilistic martingales were already considered by Buss and Minnes [25]. However, the applicability of their results for our purpose is limited. In particular, they studied two cases: probabilistic martingales which are total almost surely and probabilistic martingales which may be partial but nevertheless almost surely succeed on a given sequence. It is fairly easy to show that these cases reduce to computable and partial computable martingales respectively. The results presented in this chapter are different and require more involved proofs. Moreover, these results have been published in [14].

5.1 Preliminaries

We have already discussed in the previous chapter that algorithmic randomness' goal is to assign a meaning to the notion of individual random string or sequence. While for strings we cannot reasonably hope for a clear separation between random and non-random (instead we have a quantitative measure of randomness: Kolmogorov complexity, see section 4.1.1), for infinite binary sequences one can get such a separation. There are in fact many possible definitions. The most important one is called Martin-Löf randomness: the reason Martin-Löf's definition of randomness is considered to be the central one is that it is both well-behaved (Martin-Löf random sequences possess most properties one would expect from 'random' sequences, including computability-theoretic properties) and robust, in that one can naturally get to the same notion by different approaches (which we have already discussed in section 4.1.3). As discussed above a natural paradigm to define randomness is via unpredictability. In the next section, we review this approach and two related notions of effective randomness, other than ML-randomness.

5.1.1 Computable and partial computable randomness

According to the unpredictability approach, we want to say that a sequence X is random if its bits cannot be guessed with better-than-average accuracy. This is formalized via the notion of martingale, which we have already introduced in Definition 4.1.9. For convenience, we recall here the related terminology.

A function $d : 2^{<\mathbb{N}} \to \mathbb{R}^{>0}$ is called a *martingale* if for all $\sigma \in 2^{<\mathbb{N}}$:

$$d(\sigma) = \frac{d(\sigma 0) + d(\sigma 1)}{2}$$

Moreover, we say that a martingale d *succeeds* on a sequence X if

$$\limsup_{n\to\infty} d(X \upharpoonright n) = \infty.$$

A martingale represents the outcome of a gambling strategy in a fair game where the gambler guesses bits one by one by betting some amount of money at each stage, doubling the stake if correct, losing the stake otherwise, debts not being allowed. The quantity $d(\sigma)$ represents the capital of the gambler after having seen σ. Usually in the literature martingales are allowed to take value 0 but not allowing it makes no difference for the definitions that follow and avoids some pathological cases later in the chapter.

Armed with the notion of martingale, we can now formulate an important definition of "randomness", known as *computable randomness.*

Definition 5.1.1. A sequence $X \in \mathbf{2}^{\mathbb{N}}$ is called *computably random* if no computable martingale succeeds on X.

In the above definition, we consider only martingales that are total computable. We would also like to allow partial computable martingales, but since they are not total functions in general, they are not even martingales in the above sense. To remedy this, one can simply define a partial martingale as a function d taking values in $\mathbb{R}^{>0}$ whose domain is contained in $2^{<\mathbb{N}}$ and closed under the prefix relation (if $d(\sigma)$ is defined, $d(\tau)$ is defined for every prefix τ of σ) and furthermore for every σ, $d(\sigma 0)$ is defined if and only if $d(\sigma 1)$ is defined and in case both are defined, the fairness condition $d(\sigma) = (d(\sigma 0) + d(\sigma 1))/2$ applies. Finally, success is defined in the same way as for martingales: we say that d succeeds on X if $d(X \upharpoonright n)$ is defined for all n and $\limsup_{n\to\infty} d(X \upharpoonright n) = \infty$. We can now get the following strengthening of computable randomness.

Definition 5.1.2. A sequence $X \in \mathbf{2}^{\mathbb{N}}$ is called *partial computably random* if no partial computable martingale succeeds on X.

It is well-known that partial computable randomness is strictly stronger than computable randomness, but nonetheless strictly weaker than Martin-Löf randomness (see [75]).

Computable randomness and partial computable randomness are pretty robust notions. For example, it makes no difference whether we define success as achieving unbounded capital or as having a capital that tends to infinity.

Lemma 5.1.3 (folklore, see [37])**.** *For every total (resp. partial) computable martingale d there exists a (resp. partial) computable martingale d' such that d and d' succeed on exactly the same sequences and for every $A \in \mathbf{2}^{\mathbb{N}}$ we have $\limsup_{n\to\infty} d(A \upharpoonright n) = \infty$ iff $\lim_{n\to\infty} d'(A \upharpoonright n) = \infty$. Moreover, an index for d' can be found effectively from an index for d.*

Another important fact is that instead of considering computable real-valued martingales, we can restrict ourselves to rational valued martingales that are computable as functions from $2^{<\mathbb{N}}$ to $\mathbb{Q}$ (which we sometimes refer to as *exactly computable* martingales).

Lemma 5.1.4 (Exact Computation lemma, see [83]). *For every total (resp. partial) computable martingale d, there exists a total (resp. partial) exactly computable martingale d' such that d' succeeds on every sequence on which d succeeds. Moreover, an index for d' can be effectively obtained from an index for d.*

5.1.2 Probabilistic martingales

The above definitions assume computable martingales (partial or total) are deterministic. Our goal is to understand whether probabilistic martingales (i.e., obtained by a probabilistic algorithm) can do better. Usually, to capture the idea of probabilistic algorithm, one appeals to probabilistic models of computation, such as probabilistic Turing machines. However, from a computability-theoretic perspective, where relativization to an oracle is a bread-and-butter object of study, it is equivalent to assume that an infinite sequence of random bits is drawn in advance and given as oracle to a deterministic Turing machine which then uses it as a source of randomness. Thus, we will consider *partial computable oracle martingales*, that is, Turing functionals d where for every oracle Y, d^Y (the function computed by the functional with Y given as oracle) is a partial martingale.

Definition 5.1.5. A sequence $X \in \mathbf{2}^{\mathbb{N}}$ is called a.e. computably random if for every partial computable oracle martingale d the set of oracles Y such that d^Y is a total martingale and succeeds on X has measure zero, i.e.

$$\mu\left(\left\{Y \in \mathbf{2}^{\mathbb{N}} : d^Y \text{ is total and } \limsup_{n\to\infty} d^Y(X \restriction n) = \infty\right\}\right) = 0.$$

X is said to be a.e. partial computably random if for every partial computable oracle martingale d the set of oracles Y such that d^Y succeeds on X has measure zero.

Note that we could have equivalently defined a.e. (partial) computably randomness directly from the relativization of (partial) computable randomness: a sequence X is a.e. (partial) computably random if for almost every Y, X is (partial) computably random relative to Y.

The informal question 'do probabilistic gamblers perform better than deterministic ones' can now be fully formalized by the following two questions:

- Is a.e. computable randomness equal to computable randomness?
- Is a.e. partial computable randomness equal to partial computable randomness?

In [25], Buss and Minnes studied a restricted version of this problem. They considered a model of probabilistic martingales where one further requires $d^Y(\sigma)$ to be defined for all σ and almost all Y. This is a strong restriction which allows one to use an averaging technique. If d is a probabilistic martingale with this property, it is easy to prove that the average D defined by $D(\sigma) = \int_Y d^Y(\sigma)$ is a computable martingale. If X is computably random, D fails against X, that is, there is a constant c such that $D(X \upharpoonright n) < c$ for all n. Moreover, by Fatou's lemma:

$$\int_Y \liminf_n d^Y(X \upharpoonright n) \leq \liminf_n D(X \upharpoonright n) < c \qquad (\star)$$

which in turn implies that the set $\{Y : \liminf_n d^Y(X \upharpoonright n) = \infty\}$ has measure 0. In other words, the set of Y such that d^Y strongly succeeds against X has measure 0. By Lemma 5.1.3, this means that if a sequence X is computably random if and only if for every probabilistic martingale with the Buss-Minnes condition, d fails on X with probability 1.

Our main result is that, in the general case, we no longer have an equivalence of the two models: probabilistic martingales are indeed stronger than deterministic ones.

Theorem 5.1.6. *There exist a sequence X which is partial computably random but not a.e. partial computably random and indeed not even a.e. computably random.*

We will devote the next sections to proving Theorem 5.1.6, but let us say a few words on why we believe it to be an interesting result. First of all, it is in stark contrast with Buss and Minnes' result that probabilistic martingales do not do any better than deterministic ones when they are required to be total with probability 1: in the general case, probabilistic martingales do better! Second, this is to our knowledge the first result of this kind in algorithmic randomness. If we were to define a.e. Martin-Löf randomness following the same idea (i.e., saying that X is a.e. Martin-Löf random if for almost all Y, X is Martin-Löf random relative to oracle Y), we would not get anything new, because a.e. Martin-Löf randomness coincides with Martin-Löf randomness. This is a direct consequence of the famous van Lambalgen theorem [91], which states that for every $A, B \in \mathbf{2}^{\mathbb{N}}$, the join $A \oplus B = A(0)B(0)A(1)B(1)\ldots$ is Martin-Löf random if and only if A is Martin-Löf random and B is Martin-Löf relative to A, if and only if B is Martin-Löf random and A is Martin-Löf random relative to B. Now, let X be Martin-Löf random. For almost all Y, Y is Martin-Löf random relative to X (this is simply the fact that the set of Martin-Löf random sequences has measure 1, relativized to X), thus $X \oplus Y$ is Martin-Löf random, and thus X is Martin-Löf random relative to Y. This shows that X is a.e. Martin-Löf random. We see that van Lambalgen's theorem is key in this argument. It was already known that the analogue of van Lambalgen for computable randomness fails [94], but Theorem 5.1.6 shows that it fails in a very strong sense.

Let us also remark that van Lambalgen's theorem shows that Martin-Löf randomness implies a.e. (partial) computable randomness: if X is Martin-Löf random, it is also Martin-Löf random relative to Y for almost every Y, and thus also (partial) computably random relative to Y for almost every Y.

5.2 Turing degrees of a.e.CR sequences

Before moving to the proof of Theorem 5.1.6, we give a simple degree-theoretic proof of a weaker result, namely a separation between computable randomness and a.e. computable randomness.

Recall that every Martin-Löf random sequence is computably random but a computable random sequence is not necessarily Martin-Löf random.This separation has some interesting connections with classical computability theory, as witnessed by the following theorem (recall that a sequence Y has *high Turing degree*, or simply *is high* if it computes some function $F : \mathbb{N} \to \mathbb{N}$ such that for every total computable function f, $f(n) \leq F(n)$ for almost all n).

Theorem 5.2.1 (Nies, Stephan, Terwijn [77]). *Let $Y \in \mathbf{2}^{\mathbb{N}}$. If Y computes a sequence X such that X is computably random but not Martin-Löf random, then Y has high Turing degree. Conversely, if Y has high Turing degree, then it computes some X which is computably random but not Martin-Löf random.*

It turns out that one can get an exact analogue of this theorem for a.e. computable randomness by replacing highness with a stronger notion: almost everywhere domination. A sequence Y is said to have *almost everywhere dominating Turing degree, or a.e. dominating Turing degree* if it computes an almost everywhere dominating function F, that is, a function F such that for every Turing functional Γ and almost every Z, if Γ^Z is total, then $\Gamma^Z(n) \leq F(n)$ for almost all n. See [75] for a more complete presentation of the history of this notion, originally due to Dobrinen and Simpson [35].

Theorem 5.2.2. *Let $Y \in \mathbf{2}^{\mathbb{N}}$. If Y computes a sequence X such that X is a.e. computably random but not Martin-Löf random, then Y has a.e. dominating Turing degree. Conversely, if Y has a.e. dominating Turing degree, then it computes some X which is a.e. computably random but not Martin-Löf random (in fact, it even computes some X which is a.e. computably random but not partial computably random).*

Remark 5.2.3. Nies et al.'s theorem actually states a little more than what we wrote above, namely that the sequence X in the second part of the theorem can be chosen to be Turing equivalent to Y. The analogue theorem is also true for a.e. computable randomness and a.e. domination but the proof becomes substantially more technical (we would need to introduce techniques to encode information into a computably random sequence) for only a small gain.

Proof. Let us prove the first part of the theorem by its contrapositive. Let $X \in \mathbf{2}^{\mathbb{N}}$ whose degree is not almost everywhere dominating. Suppose also X is not Martin-Löf random, i.e., $X \in \bigcap_n \mathcal{U}_n$ for $(\mathcal{U}_n)_{n\in\mathbb{N}}$ a sequence of uniformly effectively open sets with $\mu(\mathcal{U}_n) \leq 2^{-n}$. Consider the function t^X defined by $t^X(n) := \min\{s \mid X \in \mathcal{U}_n[s]\}$. Since X does not have a.e. dominating degree, there must exist a functional Γ such that

$$\mu\{Z \mid \Gamma^Z \text{is total and } \exists^\infty n\ \Gamma^Z(n) > t^X(n)\} > 0$$

When Γ^Z is total and $\Gamma^Z(n) > t^X(n)$ for infinitely many n, we have $X \in \mathcal{U}_n[\Gamma^Z(n)]$ for infinitely many n. Note that in that case $\mathcal{U}_n[\Gamma^Z(n)]$ is a clopen set which is Z-uniformly computable in Z. It is well-known that this type of test characterizes Schnorr randomness (a notion we will no discuss here but suffices to say that Schnorr randomness is weaker than computable randomness): a sequence X is Schnorr random if and only if for every computable sequence of clopen sets $\mathcal{D}_n$ such that $\mu(\mathcal{D}_n) \leq 2^{-n}$, X belongs to only finitely

$\mathcal{D}_n$ (see for example [12, Lemma 1.5.9]). Relativized to Z, this fact shows that X is not Z-Schnorr random for a positive measure of Z's, thus not Z-computably random for a positive measure of Z's.

The strategy to prove the second part of the theorem is to take the function F computed by Y and use it as a time bound on oracle martingales in order to 'totalize' them, which then allows us to use the averaging argument presented on page 116. In order for this to work, we must first prove that F can be assumed to be 'simple' (in terms of Kolmogorov complexity).

Lemma 5.2.4. *If Y has a.e. dominating Turing degree, it computes an a.e. dominating function F such that $K(F(n)) = O(\log n)$.*

Proof. Let $(\Phi_i)_{i \in \mathbb{N}}$ be an enumeration of all Turing functionals and consider the universal functional Ψ where $\Psi^{0^i 1 A} = \Phi_i^A$. It is easy to see that a function F is almost everywhere dominating if for almost all Z, either Ψ^Z is not total or $\Phi^Z(n) \leq F(n)$ for almost every n. For each Z, let $t^Z(n)$ be the minimum t, if it exists, such that $\Phi^Z(k)$ converges in time $\leq t$ for all $k \leq n$ and let $f^Z(n) = t^Z(n) + \max_{k \leq n} \Phi^Z(k)$.

Let Y be of a.e. dominating degree and $F \leq_T Y$ an almost everywhere dominating function.

For each n, let
$$\mathcal{U}_n = \{Z \mid f^Z(n) \downarrow < \infty\}$$
which is Σ^0_1 uniformly in n. We can write
$$\mathcal{U}_n = \bigcup_k \mathcal{U}_{n,k}$$
where
$$\mathcal{U}_{n,k} = \{Z \mid f^Z(n) \downarrow < k\}$$
and note that $\mathcal{U}_{n,k}$ is a clopen set, computable uniformly in n, k.

Since F is almost everywhere dominating, we have that for almost all Z and almost all n, either $f^Z(n)$ is undefined or $f^Z(n) \leq F(n)$. Said otherwise, the set
$$\mathcal{N}_0 = \limsup(\mathcal{U}_n \setminus \mathcal{U}_{n,F(n)})$$
is a nullset.

Now, for all n, let $a_n \in [0, n^2]$ be the largest integer with $\mu(\mathcal{U}_{n,F(n)}) \geq a_n/n^2$ and $F'(n)$ be the smallest k such that $\mu(\mathcal{U}_{n,k}) \geq a_n/n^2$. We see that $F'(n)$ is computable from F and furthermore,

$$K(F'(n)) \leq K(a_n) + O(1) \leq 2\log(n^2) + O(1) \leq 4\log n + O(1)$$

By definition, we have $\mu(\mathcal{U}_{n,F(n)}) \setminus \mathcal{U}_{n,F'(n)}) \leq 1/n^2$. By the Borel-Cantelli lemma,

$$\mathcal{N}_1 = \limsup(\mathcal{U}_{n,F(n)} \setminus \mathcal{U}_{n,F'(n)})$$

is a nullset. Thus, $\mathcal{N}_0 \cup \mathcal{N}_1$ is a nullset, which means that

$$\limsup(\mathcal{U}_n \setminus \mathcal{U}_{n,F'(n)})$$

is also a nullset, which in turn means that for almost all Z, for almost all n, if $f^Z(n)$ is defined, then $f^Z(n) \leq F'(n)$. By definition of f, a fortiori, for almost all Z, if Φ^Z is total, then $\Phi^Z(n) \leq F'(n)$ for almost all n. Thus the function F'

- is almost everywhere dominating
- is computable in F, hence computable in Y
- satisfies $K(F'(n)) = O(\log n)$

which finishes the proof of the lemma. □

As alluded to above, the function F is going to be used as a time bound. To see what we mean by this, consider a total (not necessarily computable) non-decreasing function $\psi : \mathbb{N} \to \mathbb{N}$. Let d be a (partial) exactly computable martingale. The time-bounded version of d with time bound ψ is the martingale d^ψ which mimics d but only allows it a time $\psi(n)$ to compute its bets on strings of length n. If d has not made a decision by this stage (either because it is in fact undefined, or because the time of computation is greater than $\psi(n)$)), the casino exclaims *"End of bets, nothing goes on the table!"* and the martingale is assumed to have placed an empty bet. Formally, $d^\psi(\lambda) = d(\lambda)$ and for any string σ and $b \in \{0, 1\}$:

$$d^\psi(\sigma b) = \begin{cases} d^\psi(\sigma) \cdot d(\sigma b)/d(\sigma) & \text{if both } d(\sigma 0)[\psi(n+1)]\downarrow \text{ and } d(\sigma 1)[\psi(n+1)]\downarrow \\ d^\psi(\sigma) \text{ otherwise} & \end{cases}$$

By definition d^ψ is always total, and when d is total, if the bound ψ dominates the convergence time of d (that is, for almost all σ, $d(\sigma)[\psi(|\sigma|)]\downarrow$), then d^ψ and d are within a multiplicative constant of one another, which in particular implies that d^ψ succeeds on the same sequences as d.

Now, let (d_i) be the effective enumeration of all (partial) exactly computable martingales with oracle. Without loss of generality, assume that d_i has a delay i imposed on it. Let F be the a.e dominating function as above. Let $\hat{d}$ be the oracle martingale defined by

$$\hat{d}^Z(\sigma) = \sum_i 2^{-i} d_i^{Z,F}(\sigma)$$

($d_i^{Z,F}$ is the time-bounded version of d_i^Z with time bound F).

It is a total martingale for every Z as all $d_i^{Z,F}$ are total martingales. Thus, its average D defined by

$$D(\sigma) = \int_Z \hat{d}^Z(\sigma)$$

is also a martingale.

Moreover, D is F- (exactly)computable. Indeed, because of the time bound F, the value of $d_i^{Z,F}(\sigma)$ only depends of the first $F(|\sigma|)$ bits of Z, and because of the delay on the d_i, only the martingales $(d_i)_{i\leq|\sigma|}$ matter in the computation of $D(\sigma)$. Thus the integral $\int_Z \hat{d}^Z(\sigma)$ is in fact a finite sum, can be computed from $F(|\sigma|)$, hence the F-computability of D. Even more precisely, the set of values $\{D(\sigma) \mid |\sigma| \leq n\}$ is computable from $F(n)$, and thus the Kolmogorov complexity of this set is at most $K(F(n)) + O(1) = O(\log n)$.

Let then X be the sequence which diagonalizes against D (that is, the sequence X constructed bit by bit where at each stage the chosen value of the next bit is the one that makes the martingale D lose money; all this will be detailed in the next section). Computing the first n bits of X only requires to know the set of values $\{D(\sigma) \mid |\sigma| \leq n\}$. Thus, we have established:

- $X \leq_T F$
- $K(X \upharpoonright n) \leq K(F(n)) + O(1) = O(\log n)$.

Since D does not succeed on X, by the exact same calculation as ($\star$) (see page 116), for almost all Z, $\hat{d}^Z$ does not succeed on X, and thus $d_i^{Z,F}$ does not succeed on X for any i.

But we also know, since F is a.e. dominating, for all i, for almost every Z, either d_i^Z is partial, or d_i^Z is total and its computation time is dominated by F, hence d^Z is within a multiplicative constant of $d^{Z,F}$.

Putting the two together, this entails that for almost all i and almost all Z, either d_i^Z is partial or it is total and does not succeed on X. In other words, X is a.e. computably random.

X has therefore all the desired properties:

- It is a.e. computably random,
- It is computable in F and thus computable in Y,
- $K(X \upharpoonright n) = O(\log n)$, ensuring that X is not only not Martin-Löf random, but not even partial computably random using a result of Merkle [65] (no partial computably random sequence can be of logarithmic complexity).

□

An important result of Binns et al. [18] is that a.e. domination is strictly stronger than highness. This gives us the promised weaker version of Theorem 5.1.6.

Corollary 5.2.5. *There exists a sequence X which is computably random but not a.e. computably random.*

Proof. Indeed, by Binn et al.'s result, take a high Turing degree $\mathbf{a}$ which is not a.e. dominating. By Theorem 5.2.1, there is an X in $\mathbf{a}$ which is computably random but not Martin-Löf random. By Theorem 5.2.2, X is not a.e. computably random either. □

5.3 The main construction

We now turn to the full proof of Theorem 5.1.6. We first recall the standard method to build a partial computably random sequence (see for example [75]). Next, we combine this construction with the so-called 'fireworks' technique which can be viewed as a probabilistic forcing to see how to defeat, with probabilistic martingales, sequences that have been built using this construction.

5.3.1 Defeating finitely many martingales

Let us begin by explaining how to construct a partial computably random sequence. Let us first consider the simple case where we are trying to defeat a single martingale d, which we assume for the moment to be total computable, by making sure its capital does not go above a certain threshold. Up to multiplying d by a small rational, we may assume that $d(\lambda) < 1$. By induction, suppose we have already built $X \upharpoonright n$ in a way that $d(X \upharpoonright i) < 1$ for all $i \leq n$. By the fairness condition, either $d((X \upharpoonright n)^\frown 0) < 1$ or $d((X \upharpoonright n)^\frown 1) < 1$. If the former is true, we set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$, otherwise we set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$. Continuing in this fashion we ensure that the martingale d does not succeed against X as its never reaches 2. Observe that when the martingale d is exactly computable, the sequence X is computable (uniformly in a code for d).

Suppose now that we have a finite family of total martingales $d_1, \dots d_n$. If we want to diagonalize against all of them at the same time, one can simply find positive rationals $q_1, \dots, q_n$ such that $\sum_{i=1}^{n} q_i \cdot d_i(\lambda) < 1$ and proceed as before against the martingale $\sum_{i=1}^{n} q_i \cdot d_i$. Again, the sequence X obtained by diagonalization against this finite family of martingales is computable uniformly in a code for the family of d_i's. But suppose now that some of the martingales in this family are partial instead of total. This does not cause much difficulty: having already built $X \upharpoonright n$, consider only the sub-family F of indices of martingales that are still defined on $(X \upharpoonright n)^\frown 0$ and $(X \upharpoonright n)^\frown 1$. The other martingales are undefined and thus will not succeed by fiat on the sequence X. Now, if $\sum_{i \in F} q_i \cdot d_i((X \upharpoonright n)^\frown 0) < 1$, set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$, otherwise set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$. Once again the sequence X defeats all of the d_i's, some of them because they become undefined at some stage, some of them because their capital never exceeds $1/q_i$. Moreover, X is still a computable sequence. It is not however computable uniformly in a code for the family of d_i's because one needs to specify which martingales become undefined in the construction and when (this is a finite amount of information but it cannot be uniformly computed) but this is not an obstacle for our purposes.

To summarize these preliminary considerations, we can make the following definition.

Definition 5.3.1. Let $(d_1, q_1), \dots (d_n, q_n)$ be a finite family where each d_i is a (code for) a partial computable martingale and q_i a positive rational. Let $\sigma \in 2^{<\mathbb{N}}$ such that, calling F the family of indices i such that $d_i(\sigma)$ converges, we have $\sum_{i \in F} q_i \cdot d_i(\sigma) < 1$. Consider the computable sequence X defined inductively by $X \upharpoonright |\sigma| = \sigma$ and if $X \upharpoonright n$ is already built, letting F_n be the family of indices such that $d_i((X \upharpoonright n)^\frown 0)$ converges, then $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$ if $\sum_{i \in F_n} q_i \cdot d_i(X \upharpoonright n)^\frown 0) < 1$ and $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$ otherwise. This sequence is called the *diagonalization against* $(d_1, q_1), \ldots, (d_n, q_n)$ *above* σ.

5.3.2 Defeating all partial computable martingales

When we have a countable family of martingales to diagonalize against, the standard way to proceed is to introduce them one by one during the game so that at any step we only have to diagonalize against a finite family as above. The delays between the introduction of martingales is flexible and therefore will be a parameter of the construction.

The diagonalizing sequence $\Delta((t_e)_{e \in N})$.

Let $(d_i)_{i \in \mathbb{N}}$ be a standard enumeration of partial computable rational valued martingales. Let $(t_e)_{e \in \mathbb{N}}$ be a family of positive integers. The sequence $\Delta((t_e)_{e \in N})$ is constructed by finite extension as follows. Start with the empty string $\sigma_0 = \lambda$ and recursively do the following. Having built σ_n, let q_{n+1} be a rational such that $\sum_{i \in F} q_i \cdot d_i(\sigma_n) < 1$ where F is the set of indices $i \in [1, n+1]$ such that $d_i(\sigma_n)$ converges. Let A be the diagonalization against $(d_1, q_1), \ldots, (d_{n+1}, q_{n+1})$ above σ_n. The sequence A is an extension of σ and is computable (see above), so let e be a code for it (say the smallest one). Define $\sigma_{n+1} = A \upharpoonright (|\sigma_n| + t_e)$. Finally, set

$$\Delta((t_e)_{e \in N}) = \bigcup_n \sigma_n$$

It is easy to check that $\Delta((t_e)_{e \in N})$ defeats all partial computable martingales. Moreover, the construction ensures the following important fact, which will be key for the rest of our proof:

Remark 5.3.2. For infinitely many e (namely, those codes that show up in the construction), the sequence $\Delta((t_e)_{e \in N})$ coincides with the computable sequence A of index e on a prefix of length $\geq t_e$.

5.3.3 Fireworks

Let $(\mathbb{P}, \leq)$ be a computable order, that is, each element $p \in \mathbb{P}$ can be encoded by a natural number and for a given pair (n, m) of natural numbers, it is decidable whether n and m are indeed codes for two elements of p and q in $\mathbb{P}$ and whether $p \leq q$. We say that a sequence $(p_i)_{i \in \mathbb{N}}$ of elements of $\mathbb{P}$ is $\mathbb{P}$-*generic* if $p_0 \geq p_1 \geq p_2 \geq \ldots$ and for every c.e. subset W of $\mathbb{P}$:

- either there exists an i such that $p_i \in W$
- or, there exists a j such that for any $q \leq p_j$, $q \notin W$

In particular, if W is dense (that is, for every $p \in \mathbb{P}$ there exists $q \leq p$ such that $q \in W$), then for every generic sequence $(p_i)_{i \in \mathbb{N}}$ there must be some i such that $p_i \in W$, in which case we say that $\mathbb{P}$ *meets* W.

For most computable orders of interest, there cannot exist a computable generic sequence. However, there is a way to probabilistically obtain one, using the so-called fireworks technique. This was first proven by Kurtz [55] who showed that one can probabilistically obtain a generic sequence when $\mathbb{P}$ is the set of strings and $\sigma \leq \tau$ when τ is a prefix of σ (Kurtz himself drew upon an argument of Martin [64] who had shown that one can probabilistically construct a hyperimmune set). The probabilistic nature of Kurtz's and Martin's arguments was somewhat hidden in their proof (they used a different framework sometimes referred to as "risking measure"). Rumyantsev and Shen [81] simplified Kurtz's presentation of this technique (although they only focused on Martin's result about hyperimmunity) by giving an explicit probabilistic algorithm. They illustrated their algorithm by a metaphor about a buyer who tries to buy fireworks in a shop, hence the name. Shen and Rumyantsev's presentation allowed Bienvenu and Patey [16, Section 1.4] to make the following generalization to any computable order.

Theorem 5.3.3 (Fireworks master theorem [16])**.** *For any computable order $\mathbb{P}$, there exists a Turing functional Φ with range $\mathbb{P}$ such that for a set of Z's of positive measure, we have that $\Phi^Z(i)$ is defined for all i and the sequence $(\Phi^Z(i))_{i \in \mathbb{N}}$ is generic.*

Proof. We sketch the proof, which is taken from [16, Section 1.4]. Fireworks technique can been seen as a sort of "probabilistic forcing" for Σ^0_1 formulas: given a computable order $\mathbb{P}$, we want to uniformly get, with positive probability, a $\mathbb{P}$-generic sequence, namely an infinite sequence $p_0 \geq p_1 \geq \ldots$ such that, for all i, the following requirement is satisfied:

$$\mathcal{R}_i \,:\, (\exists j)[p_j \in W_i \;\vee\; (\forall q \leq p_j)[q \notin W_i]],$$

where $(W_i)_{i\in\mathbb{N}}$ is a suitable list of the c.e. subsets of $\mathbb{P}$.

The algorithm for the sequence $p_0 \leq p_1 \leq \ldots$ works as follows. We start with an arbitrary $p_0 \in \mathbb{P}$. For each i, we initialize a counter $c_i = 0$ and pick a corresponding integer $n_i \in [1, N(i)]$ at random, where N is some fixed computable function. Each n_i is meant to be a cap for the counter c_i. Assume we have already built $p_0, \ldots, p_k$. Then there are two possibilities:

- $\forall q \leq p_k,\ q \notin W_i$: in this case, requirement $\mathcal{R}_i$ is automatically satisfied. We call this the Π_1 *case*.
- $\exists q \leq p_k,\ q \in W_i$: then, our strategy is to effectively search for such q and set $p_{k+1} = q$. This is called the Σ_1 *case*.

At each step k, for each i, we do the following.

- If $c_i < n_i$, we *make a passive guess*, i.e. we assume that we are in the Π_1 case. If at any later stage k' we find out that our guess is wrong (namely, there is $q \leq p_{k'}$ such that $q \in W_i$), we add 1 to the counter c_i and take another passive guess.
- Otherwise, we *make an attive guess*, namely we assume that we are in the Σ_1 case. Hence, we stop whatever we were doing for other requirements and start to search for some $q \leq p_k$ such that $q \in W_i$: if such a q is found, we let $q = p_{k+1}$.

We must now verify that the above algorithm produce a $\mathbb{P}$-generic sequence with positive probability. First, observe that there is only one possible bad case, namely that we take an incorrect active guess for some $\mathcal{R}_i$, so that the algorithm gets stuck while waiting waiting for some small enough q entering W_i.

Claim. *Assume that $\{n_j : j \neq i\}$ is fixed. Then there is at most one value of n_i for which the algorithm above gets stuck while trying to satisfy requirement $\mathcal{R}_i$.*

Proof of claim. Assume that we get stuck while trying to satisfy requirement $\mathcal{R}_i$ having chosen the value n_i. This means that we have made $n_i - 1$ incorrect passive guesses and then one incorrect active guess. Then, for any $n_i' < n_i$ we would have been fine: indeed, if the n'_i-th guess would have been an active one, we would have found our witness for $\mathcal{R}_i$, as the n'_i-th passive guess was incorrect. Moreover, any choice of $n'_i > n_i$ would also have been fine, as our n_i-th guess would then have been a passive one, and it would have been correct. ■

Thus, for any requirement $\mathcal{R}_i$, the probability to get stuck while trying to satisfy $\mathcal{R}_i$ is at most $\frac{1}{N(i)}$. Hence, the probability of success of the algorithm is at least $1 - \sum_i \frac{1}{N(i)}$, which can be made arbitrarily close to 1 by choosing a suitable N. □

It is not difficult to observe that, in fact, any 2-random (that is, $\emptyset'$-ML-random) computes, via some fireworks functional Φ, a $\mathbb{P}$-generic sequence. First notice that a fireworks argument actually gives us a uniform family of functionals: indeed, for any m, we can get a functional Φ_m which fails with probability at most 2^{-m}, by simply choosing N such that $\sum_i \frac{1}{N(i)} < 2^m$. For each m, let

$$\mathcal{U}_m = \{R : \ \Phi_m^R \text{ is either undefined or for some } i \ \mathcal{R}_i \text{ is not satisfied}\}.$$

Then $\mu(\mathcal{U}_m) < 2^m$. Moreover, each $\mathcal{U}_m$ is open, as the only case in which the algorithm Φ_m fails is when it waits in vain for a small enough q to enter W_i: but such a situation happens at some finite stage, hence having seen only a finite initial segment of R. Finally, we can check whether Φ_m is stuck at a given stage by using $\emptyset'$, as we need to check a Π^0_1 statement. Hence $(\mathcal{U}_m)_{m\in\mathbb{N}}$ is a $\emptyset'$-ML-test: thus, for every 2-random sequence Z, there is a sufficiently large m for which $Z \notin \mathcal{U}_m$, meaning that Φ_m^Z is defined and produces a $\mathbb{P}$-generic sequence.

For our proof of Theorem 5.1.6, we are going to use the order $\mathbb{P}$ whose elements are finite approximations of martingales with positive rational values. Specifically, a member of $\mathbb{P}$ is a total function f whose domain is $\{0,1\}^{\leq n}$ for some n – which we call *length of* f and denote by $lh(f)$ – whose range is $\mathbb{Q}^{>0}$, such that $f(\lambda) = 1$ and $f(\sigma) = (f(\sigma 0) + f(\sigma 1))/2$ for all σ of length $< lh(f)$. We say that $g \leq f$ if g is an extension of f (i.e., the domain of f is contained in the domain of g and the two coincide on the domain of f). It is clear that $(\mathbb{P}, \leq)$ is a computable order. It is also clear that if $f_1 \geq f_2 \geq \ldots$ is a sequence of elements of $\mathbb{P}$ such that $lh(f_i)$ tends to $+\infty$, then $D = \bigcup f_i$ is a total rational valued martingale. This is in particular the case when $(f_i)_{i\in\mathbb{N}}$ is a $\mathbb{P}$-generic sequence, because for every n, the set of elements of $\mathbb{P}$ of length at least n is dense; in this case, we say that the martingale $D = \bigcup f_i$ is a $\mathbb{P}$-*generic martingale*.

Lemma 5.3.4. *Let D be a $\mathbb{P}$-generic martingale. For every computable sequence A and integer k there exists s such that D reaches capital at least k while playing against the prefix of A of length s (that is, $D(A \upharpoonright l) > k$ for some $l < s$).*

Proof. Fix a computable A and consider the set

$$W = \{g \in \mathbb{P} \mid (\exists l)\ g(A \upharpoonright l) > k\}$$

We claim that W is a dense c.e. subset of $\mathbb{P}$. That it is c.e. is clear. Now, take any $f \in \mathbb{P}$. Let $n = lh(f)$. By definition of $\mathbb{P}$, $f(A \upharpoonright n)$ is positive, so we can pick an $m > n$ such that $2^{m-n} \cdot f(A \upharpoonright n) > k$. Let g be the martingale of length m which behaves like f up to length n and after that stage plays the doubling strategy on A (and stops betting outside of A). Formally:

$$g(\tau) = \begin{cases} f(\tau) & \text{if } |\tau| \leq n \\ f(\tau \upharpoonright n) & \text{if } |\tau| \geq n \text{ and } \tau \upharpoonright n \neq A \upharpoonright n \\ 0 & \text{if } \tau \upharpoonright n = A \upharpoonright n \text{ but } \tau \text{ is not a prefix of } A \\ f(A \upharpoonright n) \cdot 2^{|\tau|-n} & \text{if } \tau \text{ is a prefix of } A \end{cases}$$

It is easy to check that g is a finite approximation of martingale which extends f and by construction $g(A \upharpoonright m) = 2^{m-n} \cdot f(A \upharpoonright n) > k$. Thus W is indeed dense. $\square$

We can now finish the proof of our main result.

Proof of Theorem 5.1.6. By Theorem 5.3.3 applied to our partial order $(\mathbb{P}, \leq)$, there is a Turing functional Φ and a set $\mathcal{G}$ of positive measure such that for every $Z \in \mathcal{G}$, $\Phi^Z(n)$ is a $\mathbb{P}$-generic sequence. Thus for $Z \in \mathcal{G}$, $D^Z = \bigcup_n \Phi^Z(n)$ is a $\mathbb{P}$-generic martingale.

Let A be a computable sequence and e be a code for A. By Lemma 5.3.4, for every $Z \in \mathcal{G}$, there exists some l_e^Z such that D^Z – being a $\mathbb{P}$-generic martingale – reaches capital at least e at some point while playing against the prefix $A \upharpoonright l_e^Z$.

Now, for each e which is the code of a computable sequence choose some s_e large enough to have

$$\mu\{Z \in \mathcal{G} \mid l_e^Z \leq s_e\} \geq (1 - 2^{-e-1})\mu(\mathcal{G})$$

(and for e which is not a code for a computable sequence, choose s_e arbitrarily).

This guarantees that

$$\mu\{Z \in \mathcal{G} \mid (\forall e \text{ code for a computable seq.})\ l_e^Z \leq s_e\} \geq \mu(\mathcal{G})/2 > 0$$

Let $\mathcal{H}$ be the set of the left-hand side of this inequality.

Let us consider the sequence $\Delta((s_e)_{e\in\mathbb{N}})$, which by construction is partial computably random. For every $Z \in \mathcal{H}$, for every computable sequence A of code e, the martingale D^Z reaches capital at least e on $A \upharpoonright s_e$. On the other hand, by Fact 1, we know that for infinitely many e, the sequence $\Delta((s_e)_{e\in N})$ coincides with the computable sequence A of index e on a prefix of length $\geq s_e$. Thus this guarantees that for $Z \in \mathcal{H}$, D^Z reaches capital at least e while playing on $\Delta((s_e)_{e\in\mathbb{N}})$. Thus $\Delta((s_e)_{e\in\mathbb{N}})$ is partial computably random but not almost everywhere computably random since $\mathcal{H}$ has positive measure. $\square$

5.4 Conclusion and open questions

In this chapter, we have compared the power of deterministic and probabilistic prediction. To this end, we have introduced two notions—a.e. partial computable randomness and a.e. computable randomness. In contrast with Buss and Minnes' results [25], where (due to the stronger limitations on the class of martingales considered) the authors obtained equivalent characterizations of partial computable and computable randomness in terms of probabilistic martingales, our notions do not correspond to their deterministic counterparts, but are, indeed, strictly stronger. The following diagram summarizes the mutual relationships between these notions.

$$\begin{array}{ccccc} & \subsetneq & \mathrm{PCR} & \subsetneq & \\ \mathrm{a.e.PCR} & & \not\subseteq \quad \not\supseteq & & \mathrm{CR} \\ & \subsetneq & \mathrm{a.e.CR} & \subsetneq & \end{array}$$

The main results of this chapter, in fact, concern the incomparability of the notions of a.e. computable randomness and partial computable randomness: on the one hand, by Theorem 5.1.6, partial computable randomness does not imply a.e. computable randomness; on the other hand, Theorem 5.2.2 states that every a.e. dominating degree computes (actually, contains) a sequence which is a.e. computably random but not partial computably random.

We conclude this chapter by pointing out interesting further directions to be investigated on this topic.

The main goal we have achieved is the construction of a partial computable random sequence X which is not a.e. computably random: from the perspective of algorithmic randomness, this amounts to say that any sufficiently random sequence Z derandomizes X, in the sense that X is not computably random relative to Z. But how much randomness is actually needed to derandomize such a sequence? In particular, is Martin-Löf randomness enough? In this regard, we ask the following question.

Question 5.4.1. Given a partial computably random sequence X which is not a.e. computably random, can there be a Martin-Löf random sequence Z such that X is still computably random relative to Z? If so, is there always such a Z?

The second open question is more general, and strongly related with one of the main theoretical motivations leading to this work, namely the failure of the analogue of van Lambalgen's theorem for computable randomness. Theorem 5.1.6, in fact, can be regarded as a strong failure of this result for computable randomness, because of the existence of computably random sequences that, nevertheless, can be derandomized by almost every oracle. It is known that the analogue of van Lambalgen's theorem fails for other randomness notions studied in the literature, such as Schnorr randomness, Kurtz randomness and Demuth randomness (see [37]). However, we do not know if it fails in the strong sense mentioned above.

Question 5.4.2. Are there other randomness notions for which an analogue of Theorem 5.1.6 holds (namely, for which there is a random sequence which is not a.e. random)?

In particular, it seems that our constructions may be easily modified to get results about a.e. Schnorr randomness.

Bibliography

[1] U. Andrews and S. Badaev. On isomorphism classes of computably enumerable equivalence relations. *J. Symbolic Logic*, 85(1):61–86, 2020.

[2] U. Andrews, S. Badaev, and A. Sorbi. A survey on universal computably enumerable equivalence relations. In A. Day, M. Fellows, N. Greenberg, B. Khoussainov, A. Melnikov, and F. Rosamond, editors, *Computability and Complexity. Essays Dedicated to Rodney G. Downey on the Occasion of his 60th Birthday*, volume 10010 of *LNCS*, pages 418–451. Springer, Cham, 2017.

[3] U. Andrews, S. Lempp, J. S. Miller, K. M. Ng, L. San Mauro, and A. Sorbi. Universal computably enumerable equivalence relations. *J. Symbolic Logic*, 79(1):60–88, March 2014.

[4] U. Andrews and A. Sorbi. Effective inseparability, lattices, and pre-ordering relations. *Review of Symbolic Logic*, pages 1–28, 2019.

[5] U. Andrews and A. Sorbi. Joins and meets in the structure of ceers. *Computability*, 8(3-4):193–241, 2019.

[6] M. M. Arslanov. On some generalizations of a fixed point theorem. *Sov. Math.*, 25(5):1–10, 1981.

[7] G. Barmpalias, A. E. M. Lewis, and K. M. Ng. The importance of Π^0_1 classes in effective randomness. *J. Symbolic Logic*, 75(1):387–400, 2010.

[8] N. Bazhenov, M. Mustafa, L. San Mauro, A. Sorbi, and M. Yamaleev. Classifying equivalence relations in the ershov hierarchy. *Archive for Mathematical Logic*, 59(7-8):835–864, 2020.

[9] C. H. Bennett. Logical depth and physical complexity. In *The universal Turing machine: a half-century survey*, Oxford Sci. Publ., pages 227–257. Oxford Univ. Press, New York, 1988.

[10] C. Bernardi and F. Montagna. Equivalence relations induced by extensional formulae: classification by means of a new fixed point property. *Fundamenta Mathematicae*, 124(3):221–233, 1984.

[11] C. Bernardi and A. Sorbi. Classifying positive equivalence relations. *J. Symbolic Logic*, 48(3):529–538, 1983.

[12] L. Bienvenu. *Game-theoretic characterizations of randomness: unpredictability and stochasticity*. PhD thesis, Université de Provence, 2008.

[13] L. Bienvenu, V. Delle Rose, and W. Merkle. Relativized depth, 2021. Under review.

[14] L. Bienvenu, V. Delle Rose, and T. Steifer. Probabilistic vs Deterministic Gamblers. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*, volume 219 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[15] L. Bienvenu, P. Gács, M. Hoyrup, C. Rojas, and A. Shen. Algorithmic tests and randomness with respect to classes of measures. *Tr. Mat. Inst. Steklova*, 274(Algoritmicheskie Voprosy Algebry i Logiki):41–102, 2011.

[16] L. Bienvenu and L. Patey. Diagonally non-computable functions and fireworks. *Information and Computation*, 253:64–77, 2017.

[17] Laurent Bienvenu and Christopher P. Porter. Deep classes. *Bulletin of Symbolic Logic*, 22(2):249–286, 2016.

[18] S. Binns, B. Kjos-Hanssen, M. Lerman, and R. Solomon. On a conjecture of Dobrinen and Simpson concerning almost everywhere domination. *Journal of Symbolic Logic*, 71(1):119–136, 2006.

[19] L. A. Bokut' and G. P. Kukin. Undecidable algorithmic problems for semigroups, groups and rings. In *Algebra. Topology. Geometry, Vol. 25 (Russian), Itogi Nauki i Tekhniki*, pages 3–66. Akad. Nauk SSSR, Vsesoyuz. Inst. Nauchn. i Tekhn. Inform., Moscow, 1987. Translated in J. Soviet Math. **45** (1989), no. 1, 871–911.

[20] W. W. Boone. The word problem. *Ann. of Math.*, 70(2):207–265, 1959.

[21] W. W. Boone. Word problems and recursively enumerable degrees of unsolvability. A first paper on Thue systems. *Ann. of Math.*, 83(3):520–571, 1966.

[22] W. W. Boone. Word problems and recursively enumerable degrees of unsolvability. A sequel on finitely presented groups. *Ann. of Math.*, 84(1):49–84, 1966.

[23] W. W. Boone. Word problems and recursively enumerable degrees of unsolvability. An emendation. *Ann. of Math.*, 94:389–391, 1971.

[24] S. Burris and H. P. Sankappanavar. *A course in Universal Algebra*. Graduate Texts in Mathematics. Springer-Verlag, New York, Heidelberg, Berlin, 1981. The Millennium Edition, 2012 Update.

[25] S. Buss and M. Minnes. Probabilistic algorithmic randomness. *Journal of Symbolic Logic*, 78(2):579–601, 2013.

[26] C. S. Calude and A. Nies. Chaitin Ω numbers and strong reducibilities. In *Proceedings of the First Japan-New Zealand Workshop on Logic in Computer Science (Auckland, 1997)*, volume 3, pages 1162–1166, 1997.

[27] G. J. Chaitin. A theory of program size formally identical to information theory. *J. ACM*, 22(3):329–340, 1975.

[28] C. R. J. Clapham. Finitely presented groups with word problems of arbitrary degrees of insolubility. *Proceedings of the London Mathematical Society*, 3(4):633–676, 1964.

[29] A. H. Clifford and G. B. Preston. *The algebraic theory of semigroups. Vol. I.* Mathematical Surveys, No. 7. American Mathematical Society, Providence, R.I., 1961.

[30] P. M. Cohn. *Universal Algebra*, volume 6 of *Mathematics and Its Applications*. D. Reidel Publishing Company, Dordrecht, Boston, London, 2 edition, 1981.

[31] K. de Leeuw, E. Moore, C. Shannon, and N. Shapiro. Computability by probabilistic machines. In *Automata Studies*. Princeton University Press, 1956.

[32] J. C. E. Dekker. A theorem on hypersimple sets. *Proc. Amer. Math. Soc.*, 5:791–796, 1954.

[33] V. Delle Rose, L. San Mauro, and A. Sorbi. Word problems and ceers. *Math. Log. Q.*, 66(3):341–354, 2020.

[34] V. Delle Rose, L. San Mauro, and A. Sorbi. The category of equivalence relations. *Algebra and Logic*, 60(5):295–307, 2021.

[35] N. Dobrinen and S. Simpson. Almost everywhere domination. *Journal of Symbolic Logic*, 69(3):914–922, 2004.

[36] D. Doty and P. Moser. Feasible depth. In *Conference on Computability in Europe*, pages 228–237. Springer, 2007.

[37] R. G. Downey and D. R. Hirschfeldt. *Algorithmic randomness and complexity.* Theory and Applications of Computability. Springer, New York, 2010.

[38] R. G. Downey, D. R. Hirschfeldt, J. S. Miller, and A. Nies. Relativizing Chaitin's halting probability. *J. Math. Log.*, 5(2):167–192, 2005.

[39] Yu. L. Ershov. Positive equivalences. *Algebra and Logic*, 10(6):378–394, 1973.

[40] Yu. L. Ershov. *Theory of Numberings.* Nauka, Moscow, 1977. (Russian).

[41] E. Fokina, B. Khoussainov, P. Semukhin, and D. Turetsky. Linear orders realized by c.e. equivalence relations. *The Journal of Symbolic Logic*, 81(2):463–482, 2016.

[42] A. A. Fridman. Degrees of unsolvability of the problem of identity in finitely presented groups. *Soviet. Math.*, 3(Part 2):1733–1737, 1962.

[43] P. Gács. Every sequence is reducible to a random one. *Inform. and Control*, 70(2-3):186–192, 1986.

[44] P. Gács. Lecture notes on descriptional complexity and randomness. 2021.

[45] S. Gao and P. Gerdes. Computably enumerable equivalence relations. *Studia Logica*, 67(1):27–59, 2001.

[46] A. Gavruskin, S. Jain, B. Khoussainov, and F. Stephan. Graphs realised by r.e. equivalence relations. *Ann. Pure Appl. Logic*, 165(7-8):1263–1290, 2014.

[47] F. C. Hennie and R. E. Stearns. Two-tape simulations of multitape turing machines. *JACM*, 4(13):533–546, 1966.

[48] R. Hölzl, T. Kräling, and W. Merkle. Time-bounded kolmogorov complexity and solovay functions. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science 2009*, pages 392–402, 2009.

[49] J. M. Howie. *Fundamentals of semigroup theory*, volume 12 of *London Mathematical Society Monographs. New Series.* The Clarendon Press, Oxford University Press, New York, 1995. Oxford Science Publications.

[50] C. G. Jockusch, Jr. The degrees of hyperhyperimmune sets. *J. Symbolic Logic*, 34:489–493, 1969.

[51] C. G. Jockusch, Jr. and R. I. Soare. Degrees of members of Π^0_1 classes. *Pacific J. Math.*, 40:605–616, 1972.

[52] C. G. Jockusch, Jr. and R. I. Soare. Π^0_1 classes and degrees of theories. *Trans. Amer. Math. Soc.*, 173:33–56, 1972.

[53] D. W. Juedes, J. I. Lathrop, and J. H. Lutz. Computational depth and reducibility. *Theor. Comput. Sci.*, 132:37–70, 1994.

[54] B. Khoussainov. A journey to computably enumerable structures (tutorial lectures). In F. Manea, R. G. Miller, and D. Nowotka, editors, *Sailing Routes in the World of Computation, 14th Conference on Computability in Europe, CiE 2018 Kiel, Germany, July 30 August 3, 2018 Proceedings*, volume 10936 of *LNCS*, pages 1–19. Springer, Cham, 2018.

[55] S. Kurtz. *Randomness and Genericity in the Degrees of Unsolvability.* PhD thesis, University of Illinois at Urbana–Champaign, 1982.

[56] A. H. Lachlan. A note on positive equivalence relations. *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik*, 33(1):43–46, 1987.

[57] J. I. Lathrop and J. H. Lutz. Recursive computational depth. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming*, pages 132–142, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[58] M. Li and P. Vitanyi. *An introduction to Kolmogorov complexity and its applications.* Springer-Verlag, 2 edition, 1997.

[59] Z. K. Litvinceva. The complexity of individual identity problems in semigroups. *Algebra i Logika*, 9:172–199, 1970.

[60] S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1971.

[61] A.I. Mal'tsev. Sets with complete numberings. *Algebra i Logika*, 2(2):4–29, 1963. (Russian).

[62] D. A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 12:295–310, 1966.

[63] D. A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Mathematical Logic Quarterly*, 12(1):295–310, 1966.

[64] D. A. Martin. Measure, category, and degrees of unsolvability. Unpublished manuscript, 1967.

[65] W. Merkle. The complexity of stochastic sequences. *Journal of Computer and System Sciences*, 74(3):350–357, 2008.

[66] W. Merkle and N. Mihailović. On the construction of effectively random sets. *J. Symbolic Logic*, 69(3):862–878, 2004.

[67] W. Miller and D. A. Martin. The degrees of hyperimmune sets. *Mathematical Logic Quarterly*, 14:159–166, 1968.

[68] C. F. Miller III. *Group-Theoretic Decision Problems and their Classification.* Number 68 in Annals of Mathematical Studies. Princeton University Press, Princeton, New Jersey, 1971.

[69] F. Montagna. Relative precomplete numerations and arithmetic. *J. Philosphical Logic*, 11(4):419–430, 1982.

[70] F. Montagna and A. Sorbi. Universal recursion theoretic properties of r.e. preordered structures. *J. Symbolic Logic*, 50(2):397–406, 1985.

[71] P. Moser and F. Stephan. Limit-depth and DNR degrees. *Information Processing Letters*, 135:36–40, 2018.

[72] A. Myasnikov and D. Osin. Algorithmically finite groups. *J. Pure Appl. Algebra*, 215(11):2789–2796, 2011.

[73] K.M. Ng and H. Yu. On the degree structure of equivalence relations under computable reducibility. *Notre Dame J. Formal Log.*, 60:733–761, 2019.

[74] A. Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.

[75] A. Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides.* Oxford University Press, Oxford, 2009.

[76] A. Nies and A. Sorbi. Calibrating word problems of groups via the complexity of equivalence relations. *Mathematical Structures in Computer Science*, pages 1–15, 2018.

[77] A. Nies, F. Stephan, and S. Terwijn. Randomness, relativization and Turing degrees. *Journal of Symbolic Logic*, 70:515–535, 2005.

[78] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov.*, 44:143, 1955. (Russian).

[79] Moser P. and Stephan F. Depth, highness and DNR degrees. *Discret. Math. Theor. Comput. Sci.*, 19(4), 2017.

[80] M. B. Pour-El and S. Kripke. Deduction preserving "Recursive Isomorphisms" between theories. *Fund. Math.*, 61:141–163, 1967.

[81] A. Rumyantsev and A. Shen. Probabilistic constructions of computable objects and a computable version of Lovász local lemma. *Fundamenta Informaticae*, 132(1):1–14, 2014.

[82] C. Schnorr. A unified approach to the definition of random sequences. *Mathematical systems theory*, 5(3):246–258, 1971.

[83] C. Schnorr. *Zufälligkeit und Wahrscheinlichkeit*, volume 218 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin-Heidelberg-New York, 1971.

[84] V. Selivanov. Positive structures. In S. B. Cooper and S. S. Goncharov, editors, *Computability and Models*, pages 321–350. Springer, New York, 2003.

[85] V. Shavrukov. Remarks on uniformly finitely precomplete positive equivalences. *Mathematical Logic Quarterly*, 42(1):67–82, 1996.

[86] J. C. Shepherdson. Machine configuration and word problems of given degree of unsolvability. *Z. Math. Logik Grundlagen Math.*, 11:149–175, 1965.

[87] C. Smoryński. *Logical Number Theory I: An Introduction*. Springer-Verlag, Berlin, Heidelberg, 1991.

[88] R. I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

[89] R. I. Soare. *Turing computability*. Theory and Applications of Computability. Springer-Verlag, Berlin, 2016. Theory and applications.

[90] S. Tennenbaum. Degree of unsolvability and the rate of growth of functions. In *Proc. Sympos. Math. Theory of Automata (New York, 1962)*, pages 71–73. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963.

[91] M. van Lambalgen. *Random sequences*. PhD dissertation, University of Amsterdam, Amsterdam, 1987.

[92] A. Visser. Numerations, λ-calculus and arithmetic. In *To HB Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism.*, pages 259–284. Academic Press, New York, 1980.

[93] C. E. M. Yates. Recursively enumerable sets and retracing functions. *Z. Math. Logik Grundlagen Math.*, 8:331–345, 1962.

[94] L. Yu. When van Lambalgen's theorem fails. *Proceedings of the American Mathematical Society*, 135(3):861–864, 2007.