DISSERTATION

submitted

to the

Combined Faculty for the Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Nan Fang

Born in: Hubei, China

Oral examination: ....

Restricted Coding and Betting

Advisor: Privatdozent Dr. Wolfgang Merkle

Acknowledgements

This dissertation was written during my study at the Workgroup of Mathematical Logic and Theoretical Computer Science of Heidelberg University. It would not have been possible without a whole list of people.

First of all, I am grateful to my advisor, Wolfgang Merkle, not only in the scientific aspect, where he guided me in the area of my current research, provided me opportunities to attend various conferences and workshops and encouraged me to give talks there, but also in many other aspects of life, where without his enormous help my life in Heidelberg would be much more difficult.

I am also deeply indebted to my co-author George Barmpalias, with whom a large part of the work presented here were achieved. Apart from that I also benefit a lot from his understanding on many research topics and his idea about how to do research.

Moreover, I appreciate all the inspiring discussions with other members of my research group: Klaus Ambos-Spies, Nadine Losert, Martin Monath, Ivan Titov, and many other visitors to the group: Laurent Bienvenu, Rodney G. Downey, Yun Fan, Rupert Hölzl, Frank Stephan, Sebastiaan A. Terwijn, Guohua Wu, Liang Yu. I also enjoyed all the casual conversations we had, which indeed contributed to a relaxed working environment for me.

Especially, I thank Wolfgang Merkle and Martin Monath for proof-reading part of this dissertation and making lots of suggestions to improve the presentation.

I gratefully acknowledge the financial support for my whole doctoral project from China Scholarship Council.

Above all, I want to express my eternal gratitude to my parents for their love.

Abstract

One of the fundamental themes in the study of computability theory are oracle computations, i.e. the coding of one infinite binary sequence into another. A coding process where the prefixes of the coded sequence are coded such that the length difference of the coded and the coding prefix is bounded by a constant is known as cl-reducibility. This reducibility has received considerable attention over the last two decades due to its interesting degree structure and because it exhibits strong connections with algorithmic randomness. In the first part of this dissertation, we study a slightly relaxed version of cl-reducibility where the length difference is required to be bounded by some specific nondecreasing computable function h. We show that in this relaxed model some of the classical results about cl-reducibility still hold in case the function h grows slowly, at certain particular rates. Examples are the Yu-Ding theorem, which states that there is a pair of left-c.e. sequences that cannot be coded simultaneously by any left-c.e. sequence, as well as the Barmpalias-Lewis theorem that states that there is a left-c.e. sequence which cannot be coded by any random left-c.e. sequence. In case the bounding function h grows too fast, both results don't hold anymore.

Betting strategies, which can be formulated equivalently in terms of martingales, are one of the main tools in the area of algorithmic randomness. A betting strategy is usually determined by two factors, the guessed outcome at every stage and the wager on it. In the second part of this dissertation we study betting strategies where one of these factors is restricted. First we study single-sided strategies, where the guessed outcome either is always 0 or is always 1. For computable strategies we show that single-sided strategies and usual strategies have the same power for winning, whereas the latter does not hold for strongly left-c.e. strategies, which are mixtures of computable strategies, even if we extend the class of single-sided strategies to the more general class of decidably-sided strategies.

Finally, we study the case where the wagers are forced to have a certain granularity, i.e. must be multiples of some not necessarily constant betting unit. For usual strategies, wins can always be assumed to have the two following properties (a) *'win with arbitrarily small initial capital'* and (b) *'win by saving'*. In a setting of variable granularity, where the betting unit shrinks over stages, we study how the shrinking rates interact with these two properties. We show that if the granularity shrinks fast, at certain particular rates,for such granular strategies both properties are preserved. For slower rates of shrinking, we show that neither property is preserved completely, however, a weaker version of property (a) still holds. In order to investigate property (b) in this case, we consider more restricted strategies where in addition the wager is bounded from above.

Zusammenfassung

Ein zentrales Thema berechenbarkeitstheoretischer Untersuchungen sind Orakelberechnungen, d. h., die Kodierung einer unendlichen Binärfolge in einer anderen. Eine Kodierung der Präfixe der kodierten Folge derart, dass die Längendifferenz von kodiertem und kodierendem Präfix durch eine Konstante beschränkt ist, wird als cl-Reduzierbarkeit bezeichnet. Diese Reduzierbarkeit wurde wegen ihrer interessanten Gradstruktur und den engen Beziehungen zur algorithmischen Zufälligkeit in den letzten beiden Jahrzehnten intensiv untersucht. Im ersten Teil der Dissertation betrachten wir eine weniger eingeschränkte Variante der cl-Reduzierbarkeit, bei der die Längendifferenz durch eine spezielle monotone berechenbare Funktion h beschränkt ist. Wir zeigen, dass einige der klassischen Ergebnisse über die cl-Reduzierbarkeit in diesem Modell weiter gelten, falls die Funktion h nicht zu schnell, mit bestimmten Geschwindigkeiten wächst. Dies gilt zum Beispiel für den Satz von Ding und Yu, der besagt, dass es ein Paar von linksberechenbaren Folgen gibt, die nicht beide durch dieselbe linksberechenbare Folge kodiert werden können, sowie für den Satz von Barmpalias und Lewis, nach dem es eine linksberechenbare Folge gibt, die nicht durch eine zufällige linksberechenbare Folge kodiert werden kann. Beide Resultate gelten nicht mehr, falls die beschränkende Funktion h zu schnell wächst.

Wettstrategien sind eines der wichtigsten Werkzeuge des Gebiets algorithmische Zufälligkeit, diese können äquivalent durch Martingale dargestellt werden. Eine Wettstrategie wird üblicherweise durch zwei Faktoren bestimmt, den in jeder Stufe geratenen Wert und den Einsatz, der auf diesen gewettet wird. Im zweiten Teil der Dissertation untersuchen wir Wettstrategien, bei denen einer dieser Faktoren Beschränkungen unterliegt. Zuerst untersuchen wir einseitige Strategien, bei denen der geratene Wert entweder immer 0 oder immer 1 ist. Wir zeigen, dass für berechenbare Strategien einseitige Strategien und übliche Strategien auf denselben Folgen gewinnen, wohingegen dies nicht für stark linksberechenbare Strategien gilt, das sind Mischungen aus berechenbaren

Strategien, sogar dann nicht, wenn anstelle der Klasse der einseitigen Strategien die größere Klasse der effektiv-seitigen Strategien betrachtet wird.

Zuletzt betrachten wir den Fall, dass die Einsätze eine gewisse Granularität haben müssen, d. h., Vielfaches einer nicht unbedingt konstanten Einsatzeinheit sein müssen. Für übliche Strategien kann man im Fall eines Gewinns immer annehmen, dass die beiden folgenden Eigenschaften vorliegen (a) *'Gewinn bei beliebig kleinem Anfangskapital'* und (b) *'Gewinn mit Rücklagen'*. Wir untersuchen für Modelle mit variabler Granularität, bei denen der Einheitseinsatz nach und nach schrumpft, wie die Geschwindigkeit des Schrumpfens mit den beiden Eigenschaften zusammenhängt. Wir zeigen, dass beide Eigenschaften auch für granulare Strategien gelten, falls die Granularität schnell, mit bestimmten hohen Geschwindigkeiten schrumpft. Für niedrigere Geschwindigkeiten zeigen wir, dass keine der beiden Eigenschaften vollständig erhalten bleibt, jedoch gilt dann immer noch eine schwächere Version von Eigenschaft (a). Im Zusammenhang mit der Untersuchung von Eigenschaft (b) für diesen Fall betrachten wir stärker eingeschränkte Strategien bei denen der Einsatz zusätzlich nach oben beschränkt ist.

Table of contents

Chapter 1

Introduction and Notation

1.1 Introduction

Suppose we are coding one binary sequence A into another binary sequence B. There are many ways to restrict the coding process, like to limit the computational time or computational space to code the initial fragments, which are restrictions on the processing resources. Here we study restrictions on a higher level. We concern with the coding efficiency in terms of the length of the initial fragment of B required to successfully code the initial fragment of A of length n. Obviously, the longer initial fragment of B the coding process requires, the less efficient it is, in the meanwhile it might be that B is able to code more sequences because it has more space to code initial fragments of other sequences at the same stage.

If it is required that to code the initial fragment of A of length n the length of the initial fragment of B must not exceed $n + g(n)$ for some nondecreasing function g, we say it is a coding with redundancy g. In the case g is a constant, we call it coding with constant redundancy, which is also known as cl-reducibility. Over the last two decades, cl-reducibility has been intensively studied. Because on the one hand, it is a measure of relative computability, and the degrees induced by it have a different structure than the structure of Turing degrees; on the other hand, it relates to Kolmogorov complexity, which is proved to be a useful tool in the theory of algorithmic randomness.

The first part of this dissertation is devoted to studying coding with redundancy. We observe some dichotomies caused by coding with constant redundancy and with arbitrary redundancy. With these two kinds of redundancy it is indeed true that certain sequences have different coding power. As the gap between constant redundancy and arbitrary redundancy is quite large, it has been an interesting question to draw a splitting line in terms of the redundancy for these dichotomies. For one of the dichotomies, random sequences code all sequences or not, it has already been found that the splitting line lies between slow orders, which are nondecreasing functions g such that $\sum_n 2^{-g(n)}$ diverges, and fast orders, which are nondecreasing functions g such that $\sum_n 2^{-g(n)}$ converges.

In Chapter 3 we study the splitting line in terms of redundancy for another dichotomy, i.e. *one left-c.e. random sequence* codes all left-c.e. sequences or there exists one left-c.e. sequence not coded by any left-c.e. random sequences. Note that this dichotomy is stronger than the first one. Not surprisingly, we find that the splitting line also lies between slow orders and fast orders. As byproducts, we also extend some results about coding with constant redundancy to coding with small redundancy. Some of the basic concepts and notions are discussed and reviewed in Chapter 2, where the relevant results are also introduced.

In the second part of this dissertation we turn to study restricted betting strategies. In the study of algorithmic randomness, many restrictions of betting strategies have been studied. However, most of them are about the effectiveness of the strategies. Less study is about restrictions on the nature of a betting strategy, i.e. the two factors determining a betting strategy, the guessed outcome at every stage and the wager on it.

To prepare for our study about betting strategies with one of these two factors restricted, in Chapter 4 we introduce martingales, as the usual representation of betting strategies. We review some basic definitions and properties, and introduce some new notions for later use as well. Especially, we distinguish between different notions of success for a martingale. Usually a martingale succeeds (wins) on a sequence means that it can gain unbounded profit along that sequence. As a stronger notion, we say a martingale successfully saves on a sequence if it gains unbounded profit along that sequence by withdrawing them now and then to a frozen savings account.

Betting strategies with outcomes restricted are studied in Chapter 5. A strategy is 0-sided/1-sided if the guessed outcomes are always 0/1. A separable strategy is the sum of a 0-sided strategy and a 1-sided strategy. Our study shows that depending on how we formulate the effectiveness of betting strategies, this restriction may have different effects on the winning power for a class of betting strategies. If we are working with computable strategies, the restriction on the outcomes does not reduce the winning power for this strategies class. Actually, given any computable strategy there exists a separable strategy which is *superior* to the given one, in the sense that it succeeds on every sequence the given strategy succeeds on. However, if we are working with strongly left-c.e. strategies, which are mixtures of computable strategies, we see a significant difference of the winning power caused by the outcome restriction. We construct a sequence of dimension $1/2$ such that no strongly left-c.e. separable strategy wins on it. As for every sequence of dimension less than 1, there always exists a (strongly) left-c.e. strategy wins on it, this result reveals a big gap of winning power between the class of unrestricted strategies and strategies with outcome restricted. Moreover, we also study the weakly restricted decidably-sided strategies, which are strategies that the guessed outcome at every stage is computable by a total function. We also construct a sequence of dimension $1/2$ such that no strongly left-c.e. decidably-sided strategy wins on it.

Finally, in Chapter 6 we study betting strategies whose wagers are restricted, in the sense that they are forced to have some granularity, i.e. they need to be multiples of some betting unit at every stage. We know that the class of usual strategies processes two properties,

(a) **win with small initial capital:** given a strategy there exists a strategy with arbitrary small initial capital which is still superior to the given one.

(b) **win by saving:** given a strategy there exists a strategy such that it successfully saves on any sequence the given strategy succeeds on.

Under a framework of variable granularity, where the betting units shrink over stages, we find dichotomies for both properties caused by the shrinking rats. In case the granularity shrinks fast, at certain rate, in the class of such granular strategies both properties are preserved completely. In case the granularity shrinks slower, on the one hand, we show that neither property is preserved completely; on the other hand, for property (a), a weaker version is still preserved, i.e. given a strategy there exists a family of strategies with arbitrary small initial capital such that there is always at least one of them succeeds on any sequence the given strategy succeeds on. In order to investigate a weak version of property (b), we extend our research to the strategies with stronger restriction on wagers. A granular strategy is timid if its wagers at every stage are upper bounded by a constant times the granules. We found that within all timid strategies, even a weaker version of property (b) is not valid anymore. We show that there is one sequence such that some timid strategy wins on it, but no timid strategy could successfully save on it. We also reveal the weakness of timid strategies by showing that given a timid strategy there exists a family of granular strategies such that there is always at least one of them successfully saves on any sequence the given strategy succeeds on.

Basically, Chapters 3, 5 and 6 reflect the original contributions of this dissertation.

1.2 Basic Notation and Facts

Strings and sequences

A *string* usually refers to a finite binary string, which is an element of $2^{<\omega}$. While a *sequence* refers to an infinite binary sequence, which is an element of 2^{ω}.

Given a string $\sigma \in 2^{<\omega}$, its *length* is denoted by $|\sigma|$. The numbering of the bits of a string starts from 0. Hence given $n \leq |\sigma|$, the first n bits of σ, denoted by $\sigma \upharpoonright n$, are bits $\sigma(0), \ldots, \sigma(n-1)$. For an interval of natural numbers $[a, b]$ such that $b < |\sigma|$, $\sigma \upharpoonright [a, b]$ denotes the bits $\sigma(a), \ldots, \sigma(b)$. Hence we have

$$\sigma \upharpoonright n = \sigma \upharpoonright [0, n-1].$$

The same convention applies to sequences.

The *length-lexicographic ordering* on $2^{<\omega}$ is defined by saying that σ is less than τ (written $\sigma <_L \tau$) if either $|\sigma| < |\tau|$ or else both $|\sigma| = |\tau| \text{and} \sigma(n) = 0$ for the least n such that $\sigma(n) \neq \tau(n)$.

The empty string is denoted by λ. Given a nonempty string σ, σ^- denotes the string obtained by removing the last bit of σ,

$$\sigma^- = \sigma \upharpoonright (|\sigma| - 1).$$

Given two strings $\sigma, \tau \in 2^{<\omega}$, we say σ is a prefix of τ, denoted by $\sigma \preceq \tau$ (or $\tau \succeq \sigma$) if

$$\exists n \leq |\tau| \text{ such that } \sigma = \tau \upharpoonright n.$$

$\sigma \prec \tau$ if $\sigma \preceq \tau$ and $\sigma \neq \tau$. The concatenation of σ, τ is denoted by $\sigma ^\frown \tau$, i.e.

$$\sigma ^\frown \tau = \sigma(0), \dots, \sigma(|\sigma| - 1), \tau(0), \dots, \tau(|\tau| - 1).$$

A set V of strings is *prefix-free* if

$$\forall \sigma, \tau \in V \ \sigma \preceq \tau \implies \sigma = \tau.$$

Given a string $\sigma \in 2^{<\omega}$, let $n_\sigma = \text{num}(1 ^\frown \sigma) - 1$, where $\text{num}(1 ^\frown \sigma)$ is the number with binary expression $1 ^\frown \sigma$. Clearly $\sigma \mapsto n_\sigma$ is a bijection from $2^{<\omega}$ to $\mathbb{N}$.

Let $D_0 = \emptyset$. For any $n > 0$, there is unique list $\{x_i\}_{0 \leq i \leq s}$ of natural numbers such that $n = 2^{x_0} + 2^{x_1} + \cdots + 2^{x_s}$ and $x_0 < x_1 < \cdots < x_s$. Then let $D_n = \{x_i \colon 0 \leq i \leq s\}$. Clearly, $n \mapsto D_n$ is a bijection from $\mathbb{N}$ to finite subsets of $\mathbb{N}$. For the set D_n, we say it is the finite set with *canonical index* n.

With the above two bijections we can use the terms string, natural number, and finite set of natural numbers interchangeable.

We will use $\langle x_1, x_2, \cdots, x_n \rangle$ to denote an ordered n-tuple of any countable objects. For any n, there is a natural bijection from the set of all n-tuples to natural numbers. We will use this implicitly.

Sequences and reals

Binary sequences are elements of the Cantor space 2^ω. In this space, we have the following notations. The *distance* between two sequences X, Y is 2^{-n} where n is the first digit where they differ. The basic open sets in this space are

$$[\![\sigma]\!] \colon = \{X \in 2^\omega \mid \sigma \preceq X\}, \quad \sigma \in 2^{<\omega}.$$

We denote by μ the uniform measure on 2^ω. Then the measure of the basic open set $[\![\sigma]\!]$ is

$$\mu([\![\sigma]\!]) = 2^{-|\sigma|}.$$

Given a set $V \subseteq 2^{<\omega}$ we define the open set generated by V as

$$[\![V]\!] := \bigcup_{\sigma \in V} [\![\sigma]\!].$$

If V is a prefix-free set then the measure of V is

$$\mu([\![V]\!]) = \sum_{\sigma \in V} 2^{-|\sigma|}$$

If it is clear from the context, sometimes we write $\mu(V)$ instead of $\mu([\![V]\!])$.

The set of all reals is denoted by $\mathbb{R}$ and the set of all nonnegative reals is denoted by $\mathbb{R}^{0+}$.

Most of the time, a *real* refers to a binary real in the unit interval $[0, 1]$. Such a real can be regarded as a binary sequence, so we usually use them interchangeable. Following the same convention as sequences, the positions to the right of the decimal point of a real are numbered by $0, 1, 2, 3, \ldots$ from left to right. The first position to the left of the decimal point is numbered by -1.

The following notation of truncation will be used in many places. Basically, a truncation for a real (might be positive or negative) means cutting off the bits of the real beyond some position.

Notation (Truncation)**.** For a real γ and a natural number n let $[\![\gamma]\!]_n$ be the truncation of γ up to position $n-1$ of its binary expansion, that is,

$$[\![\gamma]\!]_n = \begin{cases} \lfloor \gamma \cdot 2^n \rfloor \cdot 2^{-n} & \text{if } \gamma \geq 0, \\ \lceil \gamma \cdot 2^n \rceil \cdot 2^{-n} & \text{if } \gamma < 0. \end{cases}$$

For this truncation notation it is easy to prove the following simple properties.

Proposition 1.2.1. *For $\alpha \leq \beta \in \mathbb{R}$ and $n < m \in \mathbb{N}$, the following are true.*

1. $[\![-\alpha]\!]_n = -[\![\alpha]\!]_n$.

2. *If* $\alpha \geq 0$, $[\![\alpha + \beta]\!]_n \geq [\![\alpha]\!]_n + [\![\beta]\!]_n$.

3. $[\![\alpha]\!]_n \leq [\![\beta]\!]_n$.

4. $[\![[\![\alpha]\!]_n]\!]_m = [\![[\![\alpha]\!]_m]\!]_n = [\![\alpha]\!]_n$.

5. $[\![\alpha]\!]_m - [\![\alpha]\!]_n \leq 2^{-n} - 2^{-m}$.

The following is a lemma about reals from basic analysis. We put it here for later reference.

Lemma 1.2.2. *Let* $\{a_i\}_{i\in\mathbb{N}}$ *be a series of real numbers in* $[0,1)$*. Then* $\prod_i(1-a_i) > 0$ *iff* $\sum_i a_i < \infty$.

Functions

Given a function f, the domain of f is denoted by $\mathrm{dom}(f)$. For a *partial* function $f\colon A \mapsto B$, $\mathrm{dom}(f)$ is a subset of A. If $\mathrm{dom}(f)$ equals A then it is *total.* For a function f, we let $f(x)\downarrow$ denote the statement that f is defined at x, and $f(x)\uparrow$ denote the statement that f is undefined at x.

We mainly treat two kinds of functions: functions from natural numbers to reals and functions from finite strings to reals.

Given a function $f\colon \mathbb{N} \mapsto \mathbb{R}$, we say f is *nondecreasing* (*nonincreasing*) if $f(n) \leq f(n+1)$ $(f(n) \geq f(n+1))$ for all $n \in \mathbb{N}$.

Given a function $f\colon 2^{<\omega} \mapsto \mathbb{R}$, we say f is *nondecreasing* (*nonincreasing*) if $f(\sigma^-) \leq f(\sigma)$ $(f(\sigma^-) \geq f(\sigma))$ for all $\sigma \neq \lambda$.

For two functions f and g, we write $f \leq g + \mathbf{O}(1)$ if there exists a constant c such that for all x, $f(x) \leq g(x) + c$.

Notation (Order)**.** An *order* is an unbounded nondecreasing function from $\mathbb{N}$ to $\mathbb{N}$. An order g is a *slow order* if $\sum_{n\in\mathbb{N}} 2^{-g(n)} = \infty$; and it is a *fast order* if $\sum_{n\in\mathbb{N}} 2^{-g(n)} < \infty$.

Intuitively, slow orders are the orders which grow slow. Functions bounded by $\lfloor \log n \rfloor$ belong to slow orders. While fast orders are the orders which grow fast. Functions like $(1+\epsilon)\lfloor \log n \rfloor$ for any $\epsilon > 0$ belong to fast orders. The distinction between slow orders and fast orders plays a big role in this dissertation. We will see that a large part of the main results concerns about dichotomies caused by this separation.

Logic notation

We use standard logic notation, including the following quantifiers:

- $\exists^\infty$ denotes "there exist infinitely many";
- $\forall^\infty$ denotes "for all but finitely many".

Chapter 2

Oracle Computation as Coding

This chapter is intended as a preliminary for later chapters. We discuss some of the basic notions in computability theory and algorithmic randomness, give formal definitions for the concepts which will be used later, and also point out some important or relevant results.

2.1 Effectiveness and Randomness

Effectiveness lies at the basis of the study of computability and randomness. It formulates how mathematical objects like sets of natural numbers, reals and functions, can be *realized* in a computer. One way to get effectiveness is via Turing machines, which are machines with one tape holding the inputs (in binary), one output tape, and several internal work tapes. A Turing machine reads the inputs, operates on these tapes following the instructions of a predescribed finite program and outputs the result whenever it halts. An object is considered *realizable* if there is an effective way to get information about it through a Turing machine. For example, given a function $f\colon \mathbb{N} \mapsto \mathbb{N}$, if there is a Turing machine on input any natural number n outputs the number $f(n)$, then we get every value of f though this machine directly. In this case we say the function is (Turing) computable. For another example, given a function $f\colon \mathbb{N} \mapsto \mathbb{R}$, as all reals is not a countable set, it is not realistic to find a machine output the exact values of f. However, it is possible to *approximate* the values of f by rationals. If there is a Turing machine on input any natural numbers n outputs a series $\{f_s(n)\}_{s\in\omega}$ of rationals, which is increasingly close to $f(n)$, then we can still get information about f by approximating its values arbitrarily close, though there is no way to tell how close they are. In this case we say f can be approximated from below, or it is left-c.e.

Once we have a formulation of effectiveness, we can then effectively define other notions, such as complexity and randomness. Below we review some of the formal definitions. For more details see the monographs by Odifreddi [39], Li and Vitányi [30] and Downey and Hirschfeldt [21].

Computable functions and sets

Definition 2.1.1. Let $\phi\colon \mathbb{N} \mapsto \mathbb{N}$ be a partial function. We say that ϕ is

- *partial computable* if there is a Turing machine P such that $\phi(x) = y$ iff P on input x outputs y;
- *computable* if ϕ is partial computable and the domain of ϕ is $\mathbb{N}$.

Using computable functions we can define computable sets.

Definition 2.1.2. We say a set $A \subseteq \mathbb{N}$ is

- *computably enumerable* (*c.e.*) if A is the domain of some partial computable function;
- *computable* (or *decidable*) if both A and its complement $\overline{A} = \mathbb{N} \setminus A$ are computably enumerable.

Fix an effective list of all Turing machines $\{P_e\}_{e\in\omega}$. Let Φ_e denote the partial computable function given by P_e. Let $W_e = \mathrm{dom}(\Phi_e)$, then $\{W_e\}_{e\in\omega}$ is an effective list of all c.e. sets. If $\Phi = \Phi_e$ then e is called an *index* for Φ.

A family $\{f_i\}_{i\in\omega}$ of functions is *uniformly (partial) computable* if there is a (partial) computable function f such that $f(\langle n, x\rangle) = f_n(x)$ for all n and x.

A family $\{A_i\}_{i\in\omega}$ of sets is *uniformly c.e.* if there is a c.e. set A such that $A_n = \{x\colon \langle n, x\rangle \in A\}$.

A family $\{A_i\}_{i\in\omega}$ of sets is *uniformly computable* if both $\{A_i\}_{i\in\omega}$ and $\{\overline{A_i}\}_{i\in\omega}$ are uniformly c.e.

Definition 2.1.3. We write $\Phi_{e,s}(x) = y$ if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps. We write $\Phi_{e,s}(x)\downarrow$ if there is some y such that $\Phi_{e,s}(x) = y$, and $\Phi_{e,s}(x)\uparrow$ otherwise. Further, we let $W_{e,s} = \mathrm{dom}(\Phi_{e,s})$.

A *computable enumeration* of a set A is an effective sequence $\{A_s\}_{s\in\omega}$ of finite sets such that $A_s \subseteq A_{s+1}$ for each s, and $A = \bigcup_s A_s$.

Each c.e. set W_e has the computable enumeration $\{W_{e,s}\}_{s\in\omega}$. Conversely, if A has a computable enumeration then A is c.e.

Left-c.e. reals and functions

For a real α, its *left cut* is defined as $L(\alpha) = \{q \in \mathbb{Q} : q < \alpha\}$.

Definition 2.1.4. A real α is

- *computable* if $L(\alpha)$ is computable;
- *left-c.e.* if $L(\alpha)$ is c.e. .

It is well known that a real α is left-c.e. iff there is a computable sequence of rationals $q_0 < q_1 < \cdots \to \alpha$. Such a computable sequence $\{q_i\}_{i\in\omega}$ of rationals is also called a *computable approximation* to α. Note that there is an effective list of computable approximations to all left-c.e. reals in the unit interval.

If we only care about nonnegative reals, then α is a left-c.e. real iff there is a computable sequence of nonnegative rationals $r_0, r_1, \ldots$ such that $\alpha = \sum_n r_n$.

For a family of objects (reals or functions), we use the term *mixture* to refer to the sum of them and implicitly assume that the sum converges to a finite value.

Then a left-c.e. nonnegative real is a mixture of a computable sequence of nonnegative rationals.

A sequence of reals is *uniformly computable* if the corresponding sequence of left cuts is uniformly computable, and *uniformly left-c.e.* if the corresponding sequence of left cuts is uniformly c.e.

Let D be a countable domain (such as $\mathbb{N}$, $2^{<\omega}$ or $\mathbb{Q}$).

Definition 2.1.5. A function $f\colon D \mapsto \mathbb{R}$ is

- *computable* if its values are uniformly computable reals;
- *left-c.e.* if its values are uniformly left-c.e. reals.

It can be easily checked that f is left-c.e. iff there are uniformly computable functions $f_s\colon D \mapsto \mathbb{Q}$ such that for all s and $x \in D$, we have $f_{s+1}(x) \geq f_s(x)$ and $\lim_s f_s(x) = f(x)$.

Note that there is also an effective list of computable approximations to all left-c.e. functions.

If we only care about nonnegative functions, then it is also true that f is a left-c.e. function iff there are uniformly computable functions $f_s\colon D \mapsto \mathbb{Q}$ such that for all $x \in D$, $\sum_s f_s(x) = f(x)$.

Then a left-c.e. nonnegative function is a mixture of uniformly computable nonnegative functions which take values in $\mathbb{Q}$.

Kolmogorov complexity

Let M be a Turing machine. M computes a partial computable function $2^{<\omega} \mapsto 2^{<\omega}$. We define the *$M$-complexity* of a string x as

$$\mathrm{C}_M(x) = \min\{|\sigma|\colon M(\sigma) = x\}$$

where $\min \emptyset = \infty$.

A machine R is *optimal* if for every machine M there exists a constant e_M such that

$$(\forall x)[\mathrm{C}_R(x) \leq \mathrm{C}_M(x) + e_M]$$

It is easy to show that there exists an optimal machine. Fix an optimal machine R. The (plain) Kolmogorov complexity of a string x is defined as

$$\mathrm{C}(x) = \mathrm{C}_R(x).$$

We say a machine M is *prefix-free* if its domain is a prefix-free set. In order to indicate that a machine M is prefix-free, we write $\mathrm{K}_M(x)$ instead of $\mathrm{C}_M(x)$. Similarly, A prefix-free machine S is optimal if for every prefix-free machine M there exists a constant e_M such that

$$(\forall x)[\mathrm{K}_S(x) \leq \mathrm{K}_M(x) + e_M].$$

An optimal prefix-free machine also exists.

Fix an optimal prefix-free machine U. The prefix-free Kolmogorov complexity of a string x is defined as

$$\mathrm{K}(x) = \mathrm{K}_U(x).$$

The *halting probability* of a prefix-free machine M is denoted by

$$\Omega_M = \sum_{\sigma \in \mathrm{dom}(M)} 2^{-|\sigma|}.$$

We define Ω as Ω_U. Note that Ω is a left-c.e. real.

Below is a very important theorem about prefix-free Kolmogorov complexity, known as the Kraft-Chaitin Theorem or the Machine Existence Theorem, which gives us an approach to constructing prefix-free machines.

Definition 2.1.6. A set $W = \{\langle d_i, \tau_i \rangle\}_{i\in\omega} \subset \mathbb{N} \times 2^{<\omega}$ is called a *request set*, its measure is defined as $\mu(W) = \sum_{i\in\omega} 2^{-d_i}$; if $\mu(W) \leq 1$, then it is called a *bounded request set* or *KC set*.

Theorem 2.1.7 (KC Theorem, Levin [29], Schnorr [44], Chaitin [18])**.** *Given a bounded request set $W = \{\langle d_i, \tau_i \rangle\}_{i\in\omega}$, there is a prefix-free machine M and strings σ_i of length d_i such that $M(\sigma_i) = \tau_i$ for all i and* $\mathrm{dom}(M) = \{\sigma_i \colon i \in \omega\}$*. Furthermore, an index for M can be obtained effectively from an index for W.*

Corollary 2.1.8. *Let $W = \{\langle d_i, \tau_i \rangle\}_{i\in\omega}$ be a KC set. Then*

$$\mathrm{K}(\tau_i) \leq d_i + \mathbf{O}(1).$$

Martin-Löf randomness

A subset C of 2^ω is a *c.e. class* if there is a c.e. set W such that $C = [\![W]\!]$. A family $\{C_i\}_{i\in\omega}$ of subsets of 2^ω is *uniformly c.e.* if there is a family $\{W_i\}_{i\in\omega}$ of uniformly c.e. sets such that $C_i = [\![W_i]\!]$ for all i.

Definition 2.1.9 (Martin-Löf [35])**.**

1. A *Martin-Löf test* is a family $\{U_i\}_{i\in\omega}$ of uniformly c.e. classes such that $\mu(U_i) \leq 2^{-i}$ for all i.

2. A sequence $X \in 2^\omega$ fails a Martin-Löf test$\{U_i\}_{i\in\omega}$ if $X \in \bigcap_i U_i$, otherwise X passes the test.

3. A sequence X is *Martin-Löf random* if X passes every Martin-Löf test.

By this definition a sequence is random if there is no effective way to "catch" it. And the effective way applied here is the Martin-Löf test.

The class of all Martin-Löf random sequences is denoted by MLR.

The following theorem shows that this notion of randomness can also be characterized by prefix-free Kolmogorov complexity.

Theorem 2.1.10 (Schnorr, see [18])**.** *A sequence X is Martin-Löf-random iff*

$$\mathrm{K}(X \upharpoonright n) \geq n - \mathbf{O}(1)\,.$$

An example of Martin-Löf random real is Ω, i.e. the halting probability of an optimal prefix-free machine.

Theorem 2.1.11 (Chaitin [18])**.** $\mathrm{K}(\Omega \upharpoonright n) \geq n - \mathbf{O}(1)$.

There is yet another characterization of Martin-Löf random sequences by tests.

Definition 2.1.12 (Solovay [47])**.** A *Solovay test* is a sequence $\{S_i\}_{i\in\omega}$ of uniformly c.e. sets such that $\sum_i \mu(S_i) < \infty$.

Theorem 2.1.13 (Solovay [47])**.** *A sequence X is Martin-Löf random iff for every Solovay test $\{S_i\}_{i\in\omega}$, only finitely many S_i contains a prefix of X.*

2.2 Oracle Computation and Redundancy

With effectiveness, now we can do effective coding. Suppose there is a Turing machine with an extra oracle tape, which contains the whole information of a sequence A, and during the computation the machine can always consult information on the oracle tape. If this machine on input any natural number outputs a bit 0 or 1, then it produces a sequence B. In this case, sequence A Turing computes sequence B, or we say B is effectively coded by A. In this section we review some definition and facts about oracle computation.

We extend the former definition of Turing machines to oracle Turing machines. The effective list $\{\Phi_e\}_{e\in\mathbb{N}}$ is now regarded as a list of partial functions depending on two arguments, the oracle set and the input. We write $\Phi_e^Y(x)\downarrow$ if the program P_e halts when the oracle is Y and the input is x; we write $\Phi_e^Y(x)$, or $\Phi_e(Y;x)$ for this output. $\Phi_e^Y(x)\uparrow$ stands for the negation of $\Phi_e^Y(x)\downarrow$. We call Φ_e a *Turing functional.* Let $W_e^Y = \mathrm{dom}(\Phi_e^Y)$.

Definition 2.2.1. A total function $f\colon \mathbb{N} \mapsto \mathbb{N}$ is called *Turing reducible* to Y, or computable in Y, if there is an e such that $f = \Phi_e^Y$. We denote this by

$f \le_T Y$. For a set A, $A \le_T Y$ if the characteristic function of A is Turing reducible to Y.

The Turing reducibility gives us an equivalence relation by

$$X \equiv_T Y \leftrightarrow X \le_T Y \le_T X$$

The equivalence classes are called *Turing-degrees*, or T-degrees for short. These T-degrees form a partial order denoted by $\mathcal{D}_T$. The structure of Turing degrees is one of the most important themes in computability theory.

We also extend definition 2.1.3 about computation steps.

Definition 2.2.2. We write $\Phi^Y_{e,s}(x) = y$ if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps, with all oracle queries less than s. And we let $W^Y_{e,s} = \mathrm{dom}(\Phi^Y_{e,s})$.

It is easy to observe that a terminating oracle computation only asks finitely many oracle questions. This is often referred as the *use principle.* Hence $(\Phi^Y_{e,s})_{s\in N}$ approximates Φ^Y_e, namely,

$$\Phi^Y_e(x) = y \leftrightarrow \exists s \Phi^Y_{e,s} = y \tag{2.1}$$

Definition 2.2.3. The *use* of $\Phi^Y_e(x)$, denoted by $\phi^Y_e(x)$, is defined if $\Phi^Y_e(x)\downarrow$, in which case its value is 1+the largest oracle query asked during this computation (and 0 if no question is asked at all). Similarly, $\phi^Y_{e,s}(x)$ is 1+the largest oracle question asked up to stage s.

If $\Phi^Y_e(x)$ yields the output y and the use is at most n, for $\sigma = Y \upharpoonright n$, we write $\Phi^\sigma_e(x) = y$. Then for each set Y,

$$\Phi^Y_e(x) = y \leftrightarrow \Phi^{Y\upharpoonright u}_e(x) = y, \tag{2.2}$$

where $u = \phi^Y_e(x)$.

Note that in an oracle computation, the use function is not restricted. By restricting the use function we could get stronger reducibilities. The most well-known restricted reducibility might be *weak truth-table reducibility* (wtt-reducibility), where the use function is required to be bounded by a computable function.

The following result about wtt-reducibility, known as the Kučera-Gács Theorem, is one of the most important results in algorithmic randomness, which tells us every real is coded by a Martin-Löf random real.

Theorem 2.2.4 (Kučera [28]-Gács [26] Theorem)**.** *Every real is wtt-reducible to a Martin-Löf random real.*

The wtt-reducibility is still too weak in some case. If we set a stronger restriction on the use bound, we can get a stronger reducibility.

Definition 2.2.5 (Downey, Hirschfeldt, and LaForte [23])**.** A *computable Lipschitz* reduction (cl-reduction) is a Turing reduction where on any input n the use is always bounded by $n + c$ for some constant c.

If the additive constant for a cl-reducibility is 0, it is also called a *identity bounded Turing reduction* (ibT- reduction) by Soare [46]. The study by Ambos-Spies [1] shows that the structure of cl-degrees of c.e. sets has a big difference compared to the structure of Turing degrees of c.e. sets.

In contrast with the Kučera-Gács Theorem, Downey and Hirschfeldt [21, Theorem 9.13.2] showed that there is a real which is not cl-reducible to any random real.

In this dissertation, we are especially interested in cl-reducibility in left-c.e. reals, in particular, the following two results about it.

Theorem 2.2.6 (Yu-Ding Theorem[52])**.** *There are two left-c.e. reals such that no left-c.e. real cl-computes both of them.*

Theorem 2.2.7 (Barmpalias-Lewis Theorem[6])**.** *There is a left-c.e. real such that no left-c.e. random real cl-computes it.*

The two left-c.e. reals in the Yu-Ding Theorem is usually called a maximal pair in the cl-degrees of left-c.e. reals. Maximal pairs in the cl-degrees of left-c.e. reals have been extensively studied by Ambos-Spies et al. [2].

On the other hand, it is easy to prove that every left c.e. real is wtt-reducible to *one* left-c.e. Martin-Löf random real, i.e. Ω. That is to say, the information of every left-c.e. real is coded into Ω by some machine, as long as there is no efficiency requirement for the coding. However, it might be that to code a short prefix of the real, a very long prefix of Ω is involved. The Barmpalias-Lewis theorem says that once we require the coding to be efficient, i.e. the length difference of the coded and the coding prefix is bounded by a constant, then Ω is not as powerful as before. Also note that if every left-c.e. real is coded into

Ω, then there exists no maximal pair either. Compared with the global structure of Turing degrees, the same dichotomy caused by wtt-reducibility and cl-reducibility appears in the local structure of c.e. Turing degrees as well.

If a sequence A computes B with use function bounded by $n+h(n)$ for a non-decreasing function h, sometime it is more convenient to say that A computes (codes) B with *redundancy* h, which means that during the coding, to code the first n bits of the coded sequence apart from the first n bits, $h(n)$ many subsequent bits of the coding sequence may involved.

With this terminology, cl-reducibility is a coding process with constant redundancy.

2.3 Coding by Permitting

One way to do coding is by permitting. Suppose that we want to construct a sequence A in an approximating way, and want it to be coded by another sequence B. This can be achieved as follows. Whenever we want to change the approximation to A, before making the change we will ask for permission from B, only after a permission from B is received we do the scheduled change. In this way A is coded by B in the sense that every change of A is coded into B by some permitting mechanism.

The permitting argument is heavily used in the study of the c.e. Turing-degrees. There are many permitting types developed. One of them is the multiple permitting, which is used in the case where many permissions may be asked to code one bit of the coded sequence. Multiple permitting is usually implemented by the array non-computable degrees, which was first defined and investigated by Downey, Jockusch, and Stob [22], and then revised by Downey and Hirschfeldt [21].

Definition 2.3.1 (Downey et al. [22], Downey and Hirschfeldt [21])**.**

- A sequence of finite sets $\{F_n\}_{n\in N}$ is called a *strong array* if there is a computable function f such that $F_n = D_{f(n)}$ for every $n \in N$ where D_i denotes the finite set with canonical index i.

- A strong array $\{F_n\}_{n\in\omega}$ is a *very strong array* (v.s.a.) if

  (i) $\bigcup_{n\in\omega} F_n = N$,

(ii) $F_n \cap F_m = \emptyset$ if $n \neq m$, and

(iii) $0 < |F_n| < |F_{n+1}|$ for all $n \in \mathbb{N}$.

- A c.e. set A is *array non-computable* if there is a v.s.a. $\{F_n\}_{n\in\omega}$ such that $(\forall e)(\exists n)[W_e \cap F_n = A \cap F_n]$
- A c.e. degree $\mathbf{a}$ is *array non-computable* (a.n.c.) if there is an a.n.c. set A in $\mathbf{a}$.

Proposition 2.3.2 (Downey, Jockusch, and Stob [22])**.** *If a c.e. degree* $\mathbf{d}$ *is* array non-computable*, then for all v.s.a.* $\{F_n\}_{n\in N}$ *there is a c.e. set* $D \in \mathbf{d}$ *such that*

$$(\forall e)(\exists^\infty n)[W_e \cap F_n = D \cap F_n].$$

With the above property, a multiple permitting argument by an array non-computable degree may work as follows. Assume that we want to construct (in an approximating way) a sequence A satisfying a list of requirements $\mathcal{R}_i$ and at the same time code any change of A into some sequence B in a array non-computable degree $\mathbf{d}$. First, we make infinitely many copies $\{\mathcal{Q}_{\langle i,j\rangle}\}_{j\in\omega}$ of every requirement $\mathcal{R}_i$ such that if one of $\{\mathcal{Q}_{\langle i,j\rangle}\}_{j\in\omega}$ is satisfied, $\mathcal{R}_i$ will also be satisfied. For every requirement $\mathcal{Q}_{\langle i,j\rangle}$, we find a block $A_{\langle i,j\rangle}$ of bits of A such that $A_{\langle i,j\rangle}$ is responsible for $\mathcal{Q}_{\langle i,j\rangle}$. Then we will fix some array in advance and assign $A_{\langle i,j\rangle}$ to the finite set F_j in the array. We will ensure that there are infinitely many F_j such that the size of F_j is larger than the permissions requested by requirement $\mathcal{Q}_{\langle i,j\rangle}$. During the construction of A we also construct a series $\{V_i\}$ of c.e. sets. By Proposition 2.3.2, there is a c.e. set $B \in \mathbf{d}$ such that

$$(\forall i)(\exists^\infty j)[V_i \cap F_j = B \cap F_j]. \tag{2.3}$$

Whenever a requirement $\mathcal{Q}_{\langle i,j\rangle}$ asks permission to change some bits in $A_{\langle i,j\rangle}$, we enumerate a new number into $V_i \cap F_j$, in the purpose of violating the relationship $V_i \cap F_j = B \cap F_j$. If at a later stage $V_i \cap F_j = B \cap F_j$ holds again, we say the permission is received. In this way every change of A in $A_{\langle i,j\rangle}$ is followed by a change of B in $B \cap F_j$. For every i, by (2.3), there will be infinitely many j such that the permissions asked by $\mathcal{Q}_{\langle i,j\rangle}$ will be all received. Thus, every requirement $\mathcal{R}_i$ will be satisfied. In the meanwhile every block $A_{\langle i,j\rangle}$ of A is coded by $B \cap F_j$.

Using this multiple permitting argument, the Yu-Ding Theorem 2.2.6 and the Barmpalias-Lewis Theorem 2.2.7 turn out to be characterizations of array non-computable degrees.

Theorem 2.3.3 (Barmpalias, Downey, and Greenberg [10])**.** *The following are equivalent for a c.e. degree* $\mathbf{d}$*:*

1. *There are two left-c.e. reals in* $\mathbf{d}$ *such that no left-c.e. real cl-computes both of them.*
2. *There is a left-c.e. real in* $\mathbf{d}$ *such that no left-c.e. random real cl-computes it.*
3. *There is a set in* $\mathbf{d}$ *such that no left-c.e. random real cl-computes it.*
4. $\mathbf{d}$ *is array non-computable.*

Chapter 3

Coding Left-C.E. Reals with Redundancy

We have already seen in § 2.2 that by the Kučera-Gács Theorem every real is coded by a Martin-Löf real with a computable redundancy. It is also known that the latter is not true for coding with constant redundancy. Computable redundancy could be very large, compared to constant redundancy. In order to find an optimal bound of the redundancy in the Kučera-Gács Theorem, Barmpalias and Lewis-Pye [8] analyzed both Kučera's and Gács's proofs, and found that the best redundancy by their proof methods respectively is $n \log n$ in Kučera's case and $\sqrt{n} \log n$ in Gács's case. Another paper by Merkle and Mihailović [37] achieved the same redundancy as Gács with a direct and simpler method. However, both redundancies turned out to be not optimal by subsequent papers, where a characteristic criterion for an optimal bound was revealed.

Theorem 3.0.1 (Barmpalias and Lewis-Pye [9]). *Given a computable fast order g, every real is coded by a Martin-Löf random real with redundancy g.*

Theorem 3.0.2 (Barmpalias, Lewis-Pye, and Teutsch [13]). *Given a computable slow order g, there exists a real which is not coded by any Martin-Löf random real with redundancy g.*

If we distinguish between *large redundancy functions*, which are fast orders, and *small redundancy functions*, which are slow orders or constant functions, then the above two theorems show that a redundancy function is sufficient for the Kučera-Gács Theorem if and only if it is a large redundancy function.

Turning to the local case of left-c.e. reals, a similar but stronger dichotomy caused by wtt-reducibility and cl-reducibility appears as well, i.e. one single left-c.e. random real codes all left-c.e. reals with computable redundancy *versus* there exists one left-c.e. real not coded by any left-c.e. random real with constant redundancy. Note that result of in the local case usually do not follow directly from the global case as the constructions are quite different. In this chapter we show that for the local case, the splitting line for this dichotomy is also between large redundancy and small redundancy. Moreover, we show that the extended versions of the Yu-Ding Theorem and the Barmpalias-Lewis Theorem with relaxed redundancy hold if and only if the redundancy function is small.

The outline of this chapter is as follows. First, in § 3.1 we show the simpler side, i.e. every left-c.e. real is coded by *the* Ω with large redundancy. Thus, the Yu-Ding Theorem and the Barmpalias-Lewis Theorem cannot be extended to large redundancy. Then in § 3.2 we explore a process which will be essential for the proofs in later sections. In § 3.3 the extended Yu-Ding Theorem with small redundancy is studied. We prove that with small redundancy there are maximal pairs of left-c.e. reals such that no left-c.e. real codes both of them, and moreover, such pairs can be found in every array non-computable c.e. degree. An analog for the extended Barmpalias-Lewis Theorem is treated in § 3.4. Finally in § 3.5 we point out that an extended version of Theorem 2.3.3 for small redundancy holds as well.

Most of the results presented here can be found in the paper by Barmpalias, Fang, and Lewis-Pye [12], and a subsequent paper which is still in preparation by Fang and Merkle [25].

3.1 Coding with Large Redundancy

Let us look at the easy case. We show that every left-c.e. real is coded by Ω with large redundancy.

Theorem 3.1.1 (Barmpalias et al. [12])**.** *Given a computable fast order g, every left-c.e. real is coded by Ω with redundancy g.*

Proof. Given a left-c.e. real $\alpha \leq 1$, let $\{\alpha_s\}_{s\in\omega}$ and $\{\Omega_s\}_{i\in\omega}$ be a computable approximation of α and Ω, respectively. We construct a Solovay test $\{S_i\}_{i\in\omega}$ as follows.

At each stage s, search for the least $n \leq s$ such that $\alpha_s(n) \neq \alpha_{s+1}(n)$.

- If such n exists, enumerate $\Omega_{s+1} \upharpoonright (n + g(n))$ into S_n;
- otherwise go to the next stage directly.

Clearly $\{S_i\}_{i\in\omega}$ is uniformly c.e. For each n, whenever n is the least number such that $\alpha_s(n) \neq \alpha_{s+1}(n)$, we have $[\![\alpha_{s+1}]\!]_{n+1} \geq [\![\alpha_s]\!]_{n+1} + 2^{-n-1}$. As $\{\alpha_s\}_{s\in\omega}$ is nondecreasing and converges to α, $\{[\![\alpha_s]\!]_{n+1}\}_{s\in\omega}$ is also nondecreasing and converges to $[\![\alpha]\!]_{n+1}$. And on the other hand, $[\![\alpha]\!]_{n+1} \leq \alpha \leq 1$. Thus the case that n is the least number such that $\alpha_s(n) \neq \alpha_{s+1}(n)$ can only happens at most 2^{n+1} many times. Then $\mu(S_n) \leq 2^{n+1} \cdot 2^{-n-g(n)} = 2^{1-g(n)}$. And $\sum_i \mu(S_i) \leq \sum_i 2^{1-g(n)} < \infty$. Hence, $\{S_i\}_{i\in\omega}$ is a Solovay test.

Since Ω is Martin-Löf random, there exists some m such that for all $n \geq m$, $\Omega \notin [\![S_n]\!]$. For all $n \geq m$, if $\alpha_s(n) \neq \alpha_{s+1}(n)$ for some s, then $\Omega_{s+1} \upharpoonright (n+g(n)) \in S_n$, which implies $\Omega \upharpoonright (n + g(n)) \neq \Omega_{s+1} \upharpoonright (n + g(n))$. Thus, if $\Omega \upharpoonright (n + g(n)) = \Omega_{s+1} \upharpoonright (n + g(n))$ for some s, it must be the case that $\alpha_s(n) = \alpha(n)$. This gives us a way to compute α from Ω with use bounded by $n + g(n)$ for all $n \geq m$. For $n < m$, this finite part of α can be coded by a coding constant given to the machine in advance. $\square$

Theorem 3.1.1 shows that when the coding redundancy is allowed to be a fast order, every left-c.e. real can be coded into Ω. Then of course there does not exist maximal pairs either.

Corollary 3.1.2. *Given a computable fast order g, there is a no maximal pair of left-c.e. reals which cannot be simultaneously coded by any left-c.e. real with redundancy g.*

Theorem 3.1.1 and Corollary 3.1.2 show that the Yu-Ding Theorem and the Barmpalias-Lewis Theorem cannot be extended to coding with large redundancy.

3.2 A Loading Process

Before we study coding left-c.e. reals with small redundancy, we study a special process for a real α within interval $[a, b)$. Our argument in later sections heavily depends on the observation here.

Definition 3.2.1. Given a left-c.e. real α with computable approximation $\{\alpha_s\}_{s\in\omega}$ and two natural numbers $a \leq b$, let $f = 2^{b-a} - 1$. If there are stages $\{t_i\}_{0\leq i\leq f}$ such that

(i) $\alpha_{t_0} \upharpoonright [a,b) = 0^{b-a}$;

(ii) for each $1 \leq i \leq f$, $t_{i-1} < t_i$ and $\alpha_{t_i} = \alpha_{t_i-1} + 2^{-b}$;

(iii) at each stage $t \in (t_0, t_f) \setminus \{t_i : 0 < i < f\}$, $\alpha_t \upharpoonright [a,b) = \alpha_{t-1} \upharpoonright [a,b)$.

then the interval of stages $(t_0, t_f]$ is called an α $[a,b)$-*load process* (with loading stages $\{t_i\}_{1\leq i\leq f}$).

Lemma 3.2.2. *Let $b \geq a \geq 0$. Given an α $[a,b)$-load process $(t_0, t_f]$ with loading stages $\{t_i\}_{1\leq i\leq f}$, the following are true.*

1. *For all $1 \leq i \leq f$, $\alpha_{t_i} \upharpoonright \overline{[a,b)} = \alpha_{t_i-1} \upharpoonright \overline{[a,b)}$.*

2. *$\alpha_{t_f} \upharpoonright [a,b) = 1^{b-a}$.*

3. *For each $n \in [a,b)$, there are 2^{n-a} many loading stages where the least changed bit of α is $\alpha(n)$.*

Proof. We proof by induction on $b - a$.

If $b - a = 0$, the lemma holds trivially.

Assume the lemma holds for $b - a = k \geq 0$. Now suppose $b - a = k + 1$, and $(t_0, t_f]$ is an α $[a,b)$-load process with loading stages $\{t_i\}_{1\leq i\leq f}$.

Let $f' = 2^k - 1$, then it is easy to check that $(t_0, t_{f'}]$ is an α $[a+1,b)$-load process with loading stages $\{t_i\}_{1\leq i\leq f'}$. By induction hypothesis, the following are true.

i) For all $1 \leq i \leq f'$, $\alpha_{t_i} \upharpoonright \overline{[a+1,b)} = \alpha_{t_i-1} \upharpoonright \overline{[a+1,b)}$.

ii) $\alpha_{t_{f'}} \upharpoonright [a+1,b) = 1^k$.

iii) For each $n \in [a+1,b)$, there are 2^{n-a-1} many loading stages within $[t_1, t_{f'}]$ where the least changed bit of α is $\alpha(n)$.

Moreover, by (iii) of Definition 3.2.1 and i) above we have $\alpha_{t_{f'+1}-1}(a) = \alpha_{t_{f'}}(a) = \alpha_{t_0}(a) = 0$. That is to say, $\alpha_{t_{f'+1}-1} \upharpoonright [a,b) = 0^\frown 1^k$. Then as $\alpha_{t_{f'+1}} = \alpha_{t_{f'+1}-1} + 2^{-b}$, we have the following.

iv) $\alpha_{t_{f'+1}} \upharpoonright [a, b) = 1^\frown 0^k$.

v) $\alpha_{t_{f'+1}} \upharpoonright \overline{[a, b)} = \alpha_{t_{f'+1}-1} \upharpoonright \overline{[a, b)}$

Thus, it is easy to check that $(t_{f'+1}, t_f]$ is again an α $[a+1, b)$-load process with loading stages $\{t_i\}_{f'+2 \leq i \leq f}$. Again by induction hypothesis, the following are true.

vi) For all $f' + 2 \leq i \leq f$, $\alpha_{t_i} \upharpoonright \overline{[a+1, b)} = \alpha_{t_i - 1} \upharpoonright \overline{[a+1, b)}$.

vii) $\alpha_{t_f} \upharpoonright [a+1, b) = 1^k$.

viii) For each $n \in [a+1, b)$, there are 2^{n-a-1} many loading stages within $[t_{f'+2}, t_f]$ where the least changed bit of α is $\alpha(n)$.

Then the following observations complete the induction proof.

- By i), v) and vi) we conclude that for all $1 \leq i \leq f$, $\alpha_{t_i} \upharpoonright \overline{[a, b)} = \alpha_{t_i - 1} \upharpoonright \overline{[a, b)}$.
- By (iii) of Definition 3.2.1 and vi) above we have $\alpha_{t_f}(a) = \alpha_{t_{f'+1}}(a) = 1$. Then with vii) we have $\alpha_{t_f} \upharpoonright [a, b) = 1^{k+1} = 1^{b-a}$.
- Noted that at stage $t_{f'+1}$ the least changed bit of α is $\alpha(a)$, while at all other loading stages within $[t_1, t_f]$ $\alpha(a)$ remains unchanged. Thus, there is only one loading stage within $[t_1, t_f]$ where the least changed bit of α is $\alpha(a)$. Then given iii) and viii), for each $n \in [a+1, b)$, there are $2^{n-a-1} + 2^{n-a-1} = 2^{n-a}$ many loading stages within $[t_1, t_f]$ where the least changed bit of α is $\alpha(n)$.

□

The main idea of Lemma 3.2.2 is that during the process of repeatedly adding some weight to a real α, lower positions of α will also be changed, in a controlled way.

3.3 A Maximal Pair in Coding with Small Redundancy

Now we study the extended Yu-Ding Theorem with small redundancy. The following theorem is originally proved by Barmpalias, Fang, and Lewis-Pye [12], and then Fang and Merkle [25] provide a simpler proof. Here we present the later proof.

Theorem 3.3.1 (Barmpalias, Fang, and Lewis-Pye [12];Fang and Merkle [25])**.** *Let g be a computable slow order or constant function. There are two left-c.e. reals such that no left-c.e. real computes both of them with redundancy g.*

Our proof idea is based on the idea of the Yu-Ding construction, where it is noticed that if γ is a left-c.e. real which computes both α and β, then alternatively adding little amount to α and β is sufficient to drive γ to be too large. Here we analysis this idea a little bit further, and get a more general framework which not only proves the above theorem, but could also be adapted to prove other results.

Given g as stated, let $h(n) = n + g(n)$.

Let $\{\Phi_i, \Psi_i, \gamma_i\}_{i\in\omega}$ be an effective enumeration of all triples of two Turing functionals with use functions bounded by h and one left-c.e. real in the unit interval. Then we construct a pair of left-c.e. reals α, β satisfying all of the following requirements,

$$\mathcal{R}_e\colon \alpha \neq \Phi_e^{\gamma_e} \vee \beta \neq \Psi_e^{\gamma_e}. \tag{3.1}$$

During the construction, the values of $\alpha, \beta, \Phi_e, \Psi_e, \gamma_e$ at stage s are denoted by $\alpha_s, \beta_s, \Phi_{e,s}, \Psi_{e,s}, \gamma_{e,s}$. We say requirement $\mathcal{R}_e$ *requires attention* at stage $s+1$ if

$$\alpha_s = \Phi_{e,s}^{\gamma_{e,s}} \wedge \beta_s = \Psi_{e,s}^{\gamma_{e,s}}. \tag{3.2}$$

Noted that the cost functions of Φ_e and Ψ_e are bounded by h, (3.2) actually implies that for all $n \leq s$

$$[\![\alpha_s]\!]_n = \left[\!\left[\Phi_{e,s}^{[\![\gamma_{e,s}]\!]_{h(n)}}\right]\!\right]_n \wedge [\![\beta_s]\!]_n = \left[\!\left[\Psi_{e,s}^{[\![\gamma_{e,s}]\!]_{h(n)}}\right]\!\right]_n. \tag{3.3}$$

Now fix a single requirement $\mathcal{R}_e$. Whenever $\mathcal{R}_e$ requires attention, we change some bit of α or β at a position no greater than n by adding some weight to it. Then if $\mathcal{R}_e$ requires attention again, γ_e must have changed some bit at a position no greater than $h(n)$.

Lemma 3.3.2. *Suppose there are integers $s_0 < t_0 \leq s_1$ such that requirement $\mathcal{R}_e$ requires attention at stages s_0+1, s_1+1 and $\alpha(n)$ (or $\beta(n)$) is changed at stages t_0. Then $\gamma_{e,s_1} \geq [\![\gamma_{e,s_0}]\!]_{h(n+1)} + 2^{-h(n+1)}$.*

Proof. Assume $\alpha(n)$ is changed at stage t_0, then

$$[\![\alpha_{s_0}]\!]_{n+1} < [\![\alpha_{t_0}]\!]_{n+1} \leq [\![\alpha_{s_1}]\!]_{n+1}. \tag{3.4}$$

On the other hand, as $\mathcal{R}_e$ requires attention at stages s_0+1, s_1+1, by (3.3) we have

$$[\![\alpha_{s_0}]\!]_{n+1} = \left[\!\!\left[\Phi_{e,s_0}^{[\![\gamma_{e,s_0}]\!]_{h(n+1)}}\right]\!\!\right]_{n+1} \quad \text{and} \quad [\![\alpha_{s_1}]\!]_{n+1} = \left[\!\!\left[\Phi_{e,s_1}^{[\![\gamma_{e,s_1}]\!]_{h(n+1)}}\right]\!\!\right]_{n+1}. \tag{3.5}$$

(3.4) and (3.5) implies $[\![\gamma_{e,s_0}]\!]_{h(n+1)} \neq [\![\gamma_{e,s_1}]\!]_{h(n+1)}$. As γ_e is left-c.e., we get

$$\gamma_{e,s_1} \geq [\![\gamma_{e,s_1}]\!]_{h(n+1)} \geq [\![\gamma_{e,s_0}]\!]_{h(n+1)} + 2^{-h(n+1)}.$$

□

Our observation in Lemma 3.2.2 indicates that a conservative strategy of adding always the minimal effective weight could be helpful to drive γ_e to grow too large. However, if we apply any $[a, b)$-load process on one real α trivially, γ_e could simply copy the behaviors of α, which would make the conservative behavior of α in vain. The good thing is that we have two reals α, β at disposal. We can add the minimal effective weight to α, β in turns to prevent γ_e from copying either one's behaviors.

Lemma 3.3.3. *Suppose there are stages $s_0 < t_0 \leq s_1 < t_1 \leq s_2$, where requirement $\mathcal{R}_e$ requires attention at stages s_0+1, s_1+1, s_2+1, $\alpha(n)$ and $\beta(n)$ are changed at stages t_0 and t_1 respectively. Then $\gamma_{e,s_2} > \gamma_{e,s_0} + 2^{-h(n+1)}$.*

Proof. Clearly, by Lemma 3.3.2 we have

$$\gamma_{e,s_1} \geq [\![\gamma_{e,s_0}]\!]_{h(n+1)} + 2^{-h(n+1)} \quad \text{and} \quad \gamma_{e,s_2} \geq [\![\gamma_{e,s_1}]\!]_{h(n+1)} + 2^{-h(n+1)}.$$

Thus,

$$\begin{aligned}
\gamma_{e,s_2} &\geq \left[\!\!\left[[\![\gamma_{e,s_0}]\!]_{h(n+1)} + 2^{-h(n+1)}\right]\!\!\right]_{h(n+1)} + 2^{-h(n+1)} \\
&= [\![\gamma_{e,s_0}]\!]_{h(n+1)} + 2^{-h(n+1)} + 2^{-h(n+1)} \\
&> \gamma_{e,s_0} + 2^{-h(n+1)}.
\end{aligned}$$

□

Our strategy for the proof of Theorem 3.3.1 is as follows. Whenever $\mathcal{R}_e$ requires attention we add the minimal effective weight according to Lemma 3.2.2 to α, β alternatively. We assign an appropriate interval I_e for each requirement $\mathcal{R}_e$ according to Lemma 3.3.3, so that if both of α and β complete an I_e-load process γ_e would be forced to be larger than 1, which contradicts our assumption that $\gamma_e \leq 1$. Moreover, we will make sure all I_e disjoint with each other. Then by the definition of an I_e-load process and Lemma 3.2.2, the actions for different requirements will work independently without interfering each other.

Proof of Theorem 3.3.1. Let $m_0 = 0$. For $e \geq 0$, let m_{e+1} be the least integer such that

$$\sum_{m_e < n \leq m_{e+1}} 2^{-g(n)} \geq 2^{1+m_e}.$$

As $\sum_{i \in \mathbb{N}} 2^{-g(i)} = \infty$, such integer can always be found. Let $q_e = 2^{m_{e+1}-m_e} - 1$.

Construction:

Let $\alpha_0 = \beta_0 = 0$ and $r_{e,0} = 0$ for all e. During the construction, for any variable if no new value is specified then its value remains the same as in previous stage.

At stage $s+1$: Find the least number $e \leq s$ such that $\mathcal{R}_e$ requires attention and $r_{e,s} < 2q_e$. If exists, we say it is an e-loading stage and let $r_{e,s+1} = r_{e,s} + 1$. Then we check and do the following.

- If $r_{e,s}$ is even, add $2^{-m_{e+1}}$ to α;
- if $r_{e,s}$ is odd, add $2^{-m_{e+1}}$ to β.

Otherwise go to the next stage directly.

Verification:

From the construction we have the following trivial observation.

Observation 3.3.3.1. For every e, there are at most $2q_e$ many e-loading stages.

Then our verification is done by the following lemmas.

Lemma 3.3.4. *$\alpha, \beta \leq 1$ and they are left-c.e. reals.*

Proof. For every e, by Observation 3.3.3.1, $2^{-m_{e+1}}$ is added to α for at most q_e many times. Thus,

$$\alpha \leq \sum_{e\in\mathbb{N}} q_e \cdot 2^{-m_{e+1}} = \sum_{e\in\mathbb{N}} (2^{-m_e} - 2^{-m_{e+1}}) \leq 2^0 = 1,$$

and α is a left-c.e. real. Similarly, the same holds for β. □

Lemma 3.3.5. *For every e, if at some stage s, $r_{e,s} = 2q_e$, then $\mathcal{R}_e$ will never require attention at any later stage.*

Proof. For a contradiction, suppose for some e, s we have $r_{e,s} = 2q_e$ and $\mathcal{R}_e$ requires attention at some stage $s' > s$. Then there are already $2q_e$ many e-loading stages before stage s'. Let $t_1 < t_2 < \cdots < t_{2f}$ be these e-loading stages, where $f = q_e$. Let $t_{2f+1} = s'$.

For some $1 \leq i \leq f$, suppose at stage t_{2i-1} the least changed bit of α within $[m_e, m_{e+1})$ is $\alpha(n)$. As α and β add the same amount alternatively, then at stage t_{2i} the least changed bit of β within $[m_e, m_{e+1})$ should be $\beta(n)$. As $\mathcal{R}_e$ requires attention at stages $t_{2i-1}, t_{2i}, t_{2i+1}$ respectively, by Lemma 3.3.3 we have

$$\gamma_{e,t_{2i+1}-1} > \gamma_{e,t_{2i-1}-1} + 2^{-h(n+1)}.$$

On the other hand, $2^{-m_{e+1}}$ is added to α, β alternatively at all e-loading stages. It is easy to check that $(0, t_{2f-1}]$ is an α $[m_e, m_{e+1})$-load process with loading stages $\{t_{2i-1}\}_{1\leq i\leq f}$ and $(0, t_{2f}]$ is a β $[m_e, m_{e+1})$-load process with loading stages $\{t_{2i}\}_{1\leq i\leq f}$. By Lemma 3.2.2 for each $n \in [m_e, m_{e+1})$, there are 2^{n-m_e} many loading stages where the least changed bit of α is $\alpha(n)$. Thus,

$$\gamma_{e,t_{2f+1}-1} - \gamma_{e,t_1-1} > \sum_{m_e\leq n<m_{e+1}} 2^{n-m_e} \cdot 2^{-h(n+1)} = 2^{-1-m_e} \cdot \sum_{m_e<n\leq m_{e+1}} 2^{n-h(n)} \geq 1.$$

Then $\gamma_e \geq \gamma_{e,t_{2f+1}-1} > 1$, which contradicts our assumption that $\gamma_e \leq 1$. □

Lemma 3.3.6. *For every e, requirement $\mathcal{R}_e$ requires attention only finitely many times.*

Proof. By Lemma 3.3.5, actually $r_{e,s} < 2q_e$ already holds when $\mathcal{R}_e$ requires attention at stage $s+1$. Thus, by construction $\mathcal{R}_e$ may require attention only at i-loading stages for $i \leq e$. As for every i there are only finitely many i-loading stages, then $\mathcal{R}_e$ requires attention only finitely many times. □

Lemma 3.3.6 implies that every requirement $\mathcal{R}_e$ requires no attention from some stage on, which means that all $\mathcal{R}_e$ is satisfied eventually. This completes the proof of Theorem 3.3.1. $\square$

Now we modify the above proof to show that actually such maximal pair of left-c.e. reals can be found in every array non-computable degree.

Theorem 3.3.7 (Barmpalias, Fang, and Lewis-Pye [12];Fang and Merkle [25]). *Let g be a computable slow order or constant function and $\mathbf{d}$ be an array non-computable c.e. degree. There are two left-c.e. reals in $\mathbf{d}$ such that no left-c.e. real computes both of them with redundancy g.*

Given g as stated, let $h(n) = n+g(n)$. Let $\mathbf{d}$ be an array non-computable c.e. degree. We construct a maximal pair of left-c.e. reals as in the proof of Theorem 3.3.1 with the extra property that they are coded by some c.e. set in $\mathbf{d}$ and each one codes that set as well. To achieve this, we use the multiple permitting argument, as explained in § 2.3.

At first, suppose we have already fixed some very strong array $\{F_n\}_{n\in\omega}$. Then there is a c.e. set $D \in \mathbf{d}$ such that $(\forall e)(\exists^\infty n)[W_e \cap F_n = D \cap F_n]$. We define requirements $\{\mathcal{R}_k\}_{k\in\omega}$ the same as (3.1). Our strategy for the proof is essentially the same as before, except that for every k instead of dealing with only one requirement $\mathcal{R}_k$ now we deal with infinitely many copies of $\mathcal{R}_k$ which are represented by requirements $\mathcal{T}_{\langle k,l\rangle}$ defined as follows for all $l \in \mathbb{N}$.

$$\mathcal{T}_{\langle k,l\rangle} : V_k \cap F_l = D \cap F_l \Rightarrow \mathcal{R}_k \text{ is satisfied},$$

where the set V_k is a c.e. set we will define during the construction do handle the permission requests from requirement $\mathcal{R}_k$.

To implement the multiple permitting argument, we define a state of being active or inactive for each requirement $\mathcal{T}_{\langle k,l\rangle}$ during the construction. At stage $s+1$, requirement $\mathcal{T}_{\langle k,l\rangle}$ *requires attention* if

$$l \geq k,\ \mathcal{T}_{\langle k,l\rangle} \text{ is active, } V_{k,s} \cap F_l = D_s \cap F_l \text{ and } \mathcal{R}_k \text{ requires attention.} \qquad (3.6)$$

As before, we assign an interval $I_{\langle k,l\rangle}$ for every requirement $\mathcal{T}_{\langle k,l\rangle}$ and make sure all of them are disjoint. Whenever a requirement $\mathcal{T}_{\langle k,l\rangle}$ requires attention, before taking any action of adding weight to α or β, we request a permission by picking some $x \in F_l$ which is not yet in V_k and enumerating it into V_k. At the same time we deactivate $\mathcal{T}_{\langle k,l\rangle}$ so that it will not require any further attention. When at a later stage x enters D, we say the permission is received. Until then we perform the scheduled action and activate $\mathcal{T}_{\langle k,l\rangle}$ again. To make sure all permissions requests can be handled, the size of the array F_l should be large enough. Clearly, if $V_k \cap F_l = D \cap F_l$ holds for some l, the permissions should always be received. In this case, $\mathcal{T}_{\langle k,l\rangle}$ is a witness for the fact that requirement $\mathcal{R}_k$ is satisfied.

Our strategy ensures that the actions by each requirement $\mathcal{T}_{\langle k,l\rangle}$ work independently without interfering each other. And $\alpha \upharpoonright I_{\langle k,l\rangle}, \beta \upharpoonright I_{\langle k,l\rangle}$ are recoverable from D. On the other hand, as we also want D to be recoverable from both α and β, we will fix a computable set J which is disjoint from every $I_{\langle k,l\rangle}$, and code the information of D into $\alpha \upharpoonright J$ and $\beta \upharpoonright J$.

Proof of Theorem 3.3.7. Let $m_0 = 0$. For $e \geq 0$, let m_{e+1} be the least integer such that

$$\sum_{1+m_e<i\leq m_{e+1}} 2^{-g(i)} \geq 2^{2+m_e}.$$

As $\sum_{i\in\mathbb{N}} 2^{-g(i)} = \infty$, such integer can always be found. Let $I_e = [m_e + 1, m_{e+1})$ and $J = \{m_e | e \in \mathbb{N}\}$. Note that J and all I_e are all disjoint and their union is the set $\mathbb{N}$. Let $q_e = 2^{m_{e+1}-m_e-1} - 1$.

We fix a very strong array $\{F_l\}_{l\in N}$ such that

$$|F_l| = 1 + |F_{l-1}| + \max\{2q_{\langle k,l\rangle} \colon k \leq l\}.$$

By Proposition 2.3.2, let $D \in \mathbf{d}$ be a c.e. set such that

$$(\forall e)(\exists^\infty n)[W_e \cap F_n = D \cap F_n] \tag{3.7}$$

Without loss of generality, we assume $D_0 = \emptyset$.

Construction:

Let $e = \langle k, l\rangle$. Set all requirements $\mathcal{T}_e$ to be active.

Let $\alpha_0 = \beta_0 = 0$, $r_{e,0} = 0$ for all e and $V_{k,0} = \emptyset$ for all k. During the construction, for any variable if no new value is specified then its value remains the same as in previous stage.

At stage $2s + 1$: For each $i \in D_{s+1} \setminus D_s$, let $\alpha(m_i) = \beta(m_i) = 1$.

Find the least number $e \leq 2s$ such that $\mathcal{T}_e$ requires attention and $r_{e,2s} < 2q_e$. If exists, we say it is an e-pending stage and set $\mathcal{T}_e$ to be inactive. Then we pick some $x \in F_l \setminus V_{k,2s}$ and let $V_{k,2s+1} = V_{k,2s} \cup \{x\}$. Otherwise go to the next stage directly.

$\boxed{\textit{At stage } 2s+2\textit{:}}$ Find the least number $e \leq 2s+1$ such that requirement $\mathcal{T}_e$ is inactive and $V_{k,2s+1} \cap F_l = D_{2s+1} \cap F_l$. If exists, we say it is an e-loading stage and set $\mathcal{T}_e$ to be active. Then we let $r_{e,2s+2} = r_{e,2s+1} + 1$, check and do the following.

- If $r_{e,2s+1}$ is even, add $2^{-m_{e+1}}$ to α;
- if $r_{e,2s+1}$ is odd, add $2^{-m_{e+1}}$ to β.

Otherwise go to the next stage directly.

$\boxed{\textbf{Verification:}}$

From the construction we have the following trivial observation.

Observation 3.3.7.1. For every e, there are at most $2q_e$ many e-pending stages and at most $2q_e$ many e-loading stages.

Let $e = \langle k, l \rangle$. Our verification is done by the following lemmas.

Lemma 3.3.8. *For each e, for all s, $|F_l \cap V_{k,s}| < |F_l|$.*

Proof. First, we notice that $|F_l \cap V_k|$ increases only at e-pending stages, and at each such stage it increases by 1.

If $l < k$, $\mathcal{T}_e$ never requires attention and there is no e-pending stage. Then for all s, $|F_l \cap V_{k,s}| = 0 < |F_l|$.

If $l \geq k$, by definition we have $|F_l| \geq 1 + 2q_e$. Then by Observation 3.3.7.1 for all s, $|F_l \cap V_{k,s}| \leq 2q_e < |F_l|$. □

Lemma 3.3.8 ensures that at every e-pending stage s, $F_l \setminus V_{k,s-1} \neq \emptyset$. Thus, every permission request is handled.

Lemma 3.3.9. $\alpha, \beta \leq 1$ *and they are left-c.e. reals.*

Proof. For every e, by Observation 3.3.7.1, $2^{-m_{e+1}}$ is added to α for at most q_e many times. On the other hand, for each e, 2^{-m_e-1} is added to α when e enters D. While as D is a c.e. set, this happens at most once for each e. Thus,

$$\alpha \leq \sum_{e\in\mathbb{N}} q_e \cdot 2^{-m_{e+1}} + \sum_{e\in\mathbb{N}} 2^{-m_e-1} = \sum_{e\in\mathbb{N}} (2^{-m_e-1} - 2^{-m_{e+1}} + 2^{-m_e-1}) \leq 2^0 = 1,$$

and α is a left-c.e. real. Similarly, the same holds for β. □

Lemma 3.3.10. *$\alpha, \beta =_T D$.*

Proof. For each e, as $2^{-m_{e+1}}$ is added to α (or β) for at most q_e many times, the actions for requirement $\mathcal{T}_e$ do not affect α (or β) outside the interval I_e. Then for any $n \in \mathbb{N}$, $D(n) = 1$ if and only if $\alpha(m_n) = 1$ if and only if $\beta(m_n) = 1$. So $D \leq_T \alpha, \beta$.

Assume now we are given D, fix some n. If $n = m_p$ for some number p, then clearly $\alpha(n) = \beta(n) = D(p)$. Otherwise, there must be some e such that $n \in I_e$. We find a stage s where $D_s \cap F_l = D \cap F_l$, then claim that $\alpha(n) = \alpha_{s_0}(n)$ and $\beta(n) = \beta_{s_0}(n)$, where $s_0 = s + 2 + \sum_{i<e} 4q_i$.

Suppose not, then there is a stage $s' > s_0$ such that $\alpha_{s_0}(n) \neq \alpha_{s'}(n)$ or $\beta_{s_0}(n) \neq \beta_{s'}(n)$. That is to say, there should be at least one e-loading stage within stages $(s_0, s']$. Let $s_1 + 1$ be the least e-loading stage within stages $(s_0, s']$. Then $V_{k,s_1} \cap F_l = D_{s_1} \cap F_l$.

If there is an e-pending stage $s_2 + 1$ within $(s, s_1]$, then

$$\begin{aligned} D_s \cap F_l \quad \subseteq \quad D_{s_2} \cap F_l \quad = \quad & V_{k,s_2} \cap F_l \\ \subsetneq \quad & V_{k,s_2+1} \cap F_l \quad = \quad V_{k,s_1} \cap F_l \quad = \quad D_{s_1} \cap F_l \quad \subseteq \quad D \cap F_l, \end{aligned}$$

which contradicts $D_s \cap F_l = D \cap F_l$.

If there is no e-pending stage within $(s, s_1]$, then $\mathcal{T}_e$ is inactive at all stages within $(s, s_1]$ and $V_{k,t} \cap F_l$ does not change for all $t \in (s, s_1]$. On the other hand, as $D_s \cap F_l = D \cap F_l$, $D_t \cap F_l$ also does not change for all $t \in (s, s_1]$. Thus,

$$V_{k,t} \cap F_l \quad = \quad V_{k,s_1} \cap F_l \quad = \quad D_{s_1} \cap F_l \quad = \quad D_t \cap F_l.$$

This implies that for all $t \in (s, s_1]$, if t is even stage t should be an i-loading stage for some $i < e$. As for each i there are at most $2q_i$ many i-loading stages, then $s_1 + 1 \leq s + 2 + \sum_{i<e} 2 \cdot 2q_i = s_0$, which contradicts the assumption $s_1 + 1 > s_0$. □

Lemma 3.3.11. *For every e, if at some stage s, $r_{e,s} = 2q_e$ then $\mathcal{T}_e$ will never require attention at any later stage.*

Proof. For a contradiction, suppose for some e, s we have $r_{e,s} = 2q_e$ and $\mathcal{T}_e$ requires attention at some stage $s' > s$. Then there are already $2q_e$ many e-

pending and $2q_e$ many e-loading stages before stage s'. Let $s_1 < s_2 < \cdots < s_{2f}$ be these e-pending stages and $t_1 < t_2 < \cdots < t_{2f}$ be these e-loading stages, where $f = q_e$. Let $s_{2f+1} = s'$. By construction it follows that $s_i < t_i < s_{i+1}$ for each $i \in [1, 2f]$.

For some $1 \le i \le f$, suppose at stage t_{2i-1} the least changed bit of $\alpha \upharpoonright I_e$ is $\alpha(n)$. As α and β add the same amount alternatively, then at stage t_{2i} the least changed bit of $\beta \upharpoonright I_e$ should be $\beta(n)$. As $\mathcal{T}_e$ requires attention at stages $s_{2i-1}, s_{2i}, s_{2i+1}$ respectively, which implies that $\mathcal{R}_k$ also requires attention at these stages, by Lemma 3.3.3 we have

$$\gamma_{k,t_{2i+1}-1} > \gamma_{k,t_{2i-1}-1} + 2^{-h(n+1)}.$$

On the other hand, $2^{-m_{e+1}}$ is added to α, β alternatively at all e-loading stages. It is easy to check that $(0, t_{2f-1}]$ is an α I_e-load process with loading stages $\{t_{2i-1}\}_{1\le i\le f}$ and $(0, t_{2f}]$ is a β I_e-load process with loading stages $\{t_{2i}\}_{1\le i\le f}$. By Lemma 3.2.2 for each $n \in I_e$, there are 2^{n-m_e-1} many loading stages where the least changed bit of α is $\alpha(n)$. Thus,

$$\gamma_{k,t_{2f+1}-1} - \gamma_{k,t_1-1} > \sum_{n\in I_e} 2^{n-m_e-1} \cdot 2^{-h(n+1)} = 2^{-m_e-2} \cdot \sum_{n\in I_e} 2^{-g(n+1)} \ge 1.$$

Then $\gamma_k \ge \gamma_{k,t_{2f+1}-1} > 1$, which contradicts our assumption that $\gamma_k \le 1$. $\square$

Lemma 3.3.12. *For every e, requirement $\mathcal{T}_e$ requires attention only finitely many times.*

Proof. By Lemma 3.3.11, actually $r_{e,s} < 2q_e$ already holds when $\mathcal{T}_e$ requires attention at stage $s+1$. Thus, by construction for odd stages, $\mathcal{T}_e$ may require attention only at i-pending stages for $i \le e$. As for every i there are only finitely many i-pending stages, then $\mathcal{T}_e$ requires attention only at finitely many odd stages, and after the last such stage it will not require attention any more. $\square$

Now fix some k. As V_k is a c.e. set, by (3.7), there are infinitely many l such that $V_k \cap F_l = D \cap F_l$. Fix some $l \ge k$ such that $V_k \cap F_l = D \cap F_l$. Then there is a stage s_0 such that $V_{k,s} \cap F_l = D_s \cap F_l$ for all $s \ge s_0$. As there are only finitely many $\langle k,l\rangle$-pending stages, there is a stage $s_1 \ge s_0$ such that there is no $\langle k,l\rangle$-pending stage after stage s_1. Then if $\mathcal{T}_{\langle k,l\rangle}$ is active at some stage $s' \ge s_1$, it will remain active thereafter, because it could only become inactive at $\langle k,l\rangle$-pending stages. If $\mathcal{T}_{\langle k,l\rangle}$ is inactive at stage s_1, by the same argument as in the proof of Lemme 3.3.10, $\mathcal{T}_{\langle k,l\rangle}$ will become active and remain active thereafter at the latest at stage $s_1 + 2 + \sum_{i<\langle k,l\rangle} 4q_i$. Moreover, by Lemma 3.3.12 there is a stage $s_2 \geq s_1 + 2 + \sum_{i<\langle k,l\rangle} 4q_i$ such that for all $s \geq s_2$, requirement $\mathcal{T}_{\langle k,l\rangle}$ does not require attention, which then implies $\mathcal{R}_k$ does not require attention. Thus, $\mathcal{R}_k$ is satisfied eventually. This completes the proof of Theorem 3.3.7. □

3.4 Random Reals Fail in Coding with Small Redundancy

Now we study the extended Barmpalias-Lewis Theorem with small redundancy. Note that constant redundancy is just a special case here, while the proof here is even simpler than the proofs shown in [6] and [10] for constant redundancy.

Theorem 3.4.1 (Fang and Merkle [25])**.** *Let g be a computable slow order or constant function. There is a left-c.e. real such that no left-c.e. random real computes it with redundancy g.*

Given g as stated, let $h(n) = n + g(n)$. Our proof idea is as follows. We construct a left-c.e. real α such that any left-c.e. real γ computing α with use function h fails to pass some Martin-Löf test.

Let $\langle \Phi_k, \gamma_k\rangle$ be an effective enumeration of all pairs of a Turing functional with use function bounded by h and a left-c.e. real in the unit interval. Then we construct a left-c.e. real α satisfying all of the following requirements,

$$\mathcal{R}_k \colon \alpha = \Phi_k^{\gamma_k} \Rightarrow \gamma_k \in \bigcap_j U_{k,j}, \tag{3.8}$$

where for each k, $\{U_{k,j}\}_{j\in\mathbb{N}}$ is some Martin-Löf test we specifically define for requirement $\mathcal{R}_k$. We break each $\mathcal{R}_k$ into

$$\mathcal{Q}_{\langle k,j\rangle} \colon \alpha = \Phi_k^{\gamma_k} \Rightarrow \gamma_k \in [\![E_{\langle k,j\rangle}]\!] \tag{3.9}$$

where $\{E_{\langle k,j\rangle}\}$ are uniformly c.e. sets we will enumerate during the construction. The idea is that each $[\![E_{\langle k,j\rangle}]\!]$ serves as the jth member of the Martin-Löf test we define for $\mathcal{R}_k$, so we have the following restriction on them,

$$(\forall k)(\forall j)\ [\mu(E_{\langle k,j\rangle}) \leq 2^{-j}]. \tag{3.10}$$

Clearly, if all requirements $\mathcal{Q}_{\langle k,j\rangle}$ are satisfied then all requirements $\mathcal{R}_k$ are also satisfied.

Let $e = \langle k, j\rangle$. During the construction, the values of $\alpha, \Phi_k, \gamma_k, E_e$ at stage s are denoted by $\alpha_s, \Phi_{k,s}, \gamma_{k,s}, E_{e,s}$. Without loss of generality, we assume that

$$(\forall s)\,[\gamma_{k,s} = [\![\gamma_{k,s}]\!]_s]. \tag{3.11}$$

We say requirement $\mathcal{Q}_e$ *requires attention* at stage $s+1$ if

$$\alpha_s = \Phi_{k,s}^{\gamma_{k,s}} \wedge \gamma_{k,s} \notin [\![E_{e,s}]\!]. \tag{3.12}$$

Noted that the cost function of Φ_k is bounded by h, (3.12) actually implies that for all $n < s$

$$[\![\alpha_s]\!]_n = \left[\!\!\left[\Phi_{k,s}^{[\![\gamma_{k,s}]\!]_{h(n)}}\right]\!\!\right]_n \wedge \gamma_{k,s} \notin [\![E_{e,s}]\!]. \tag{3.13}$$

Now fix a single requirement $\mathcal{Q}_e$. The same as before, we use our observation in Lemma 3.2.2 to drive γ_k to grow too large. Instead of having two reals α, β at disposal, now we have a left-c.e. real α and a serials of c.e. sets $\{E_e\}_{e\in\omega}$. We want to apply some $[a, b)$-load process on the real α, while prevent γ_k from copying the behaviors of α. With the help of the sets $\{E_e\}_{e\in\omega}$, this is still possible.

We define a set

$$G(e, s, m) = \{\sigma\colon [|\sigma| = s] \wedge [\gamma_{k,s} \upharpoonright s \leq_L \sigma <_L (\gamma_{k,s} + 2^{-m}) \upharpoonright s]\}.$$

Lemma 3.4.2. *If there is a stage s such that $\gamma_{k,s} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}$ and $E_{e,s+1} = E_{e,s} \cup G(e, s, h(n)+j+1)$ then $\mu(E_{e,s+1}) - \mu(E_{e,s}) \leq 2^{-h(n)-j-1}$; and if further, $\mathcal{Q}_e$ requires attention at stage $t+1$ where $t > s$, then $\gamma_{k,t} \geq [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}$.*

Proof. As $\gamma_{k,s} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}$ and $\gamma_{k,s} = [\![\gamma_{k,s}]\!]_s$, it must hold that $h(n) + j + 1 \leq s$. Then there are $2^{s-h(n)-j-1}$ many strings in the set $G(e, s, h(n)+j+1)$. Thus,

$$\mu(E_{e,s+1}) - \mu(E_{e,s}) \leq \mu(G(e, s, h(n)+j+1)) = 2^{s-h(n)-j-1} \cdot 2^{-s} = 2^{-h(n)-j-1}.$$

If requirement $\mathcal{Q}_e$ requires attention at stages $t+1$, then $\gamma_{k,t} \notin [\![E_{e,t}]\!]$. While $E_{e,t} \supseteq E_{e,s+1} = E_{e,s} \cup G(e, s, h(n)+j+1)$, then $\gamma_{k,t} \notin [\![G(e, s, h(n)+j+1)]\!]$, i.e. $\gamma_{k,t} \upharpoonright s \notin G(e, s, h(n) + j + 1)$. As $\gamma_{k,t} \upharpoonright s \geq_L \gamma_{k,s} \upharpoonright s$, then $\gamma_{k,t} \upharpoonright s \geq_L (\gamma_{k,s} + 2^{-h(n)-j-1}) \upharpoonright s$. Thus,

$$\gamma_{k,t} \geq [\![\gamma_{k,t}]\!]_s \geq \left[\!\left[\gamma_{k,s} + 2^{-h(n)-j-1}\right]\!\right]_s = \gamma_{k,s} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}.$$

□

Lemma 3.4.3. *If there are integers $s_1 < t_1 \leq s_2$ such that requirement $\mathcal{Q}_e$ requires attention at stages $s_1 + 1, s_2 + 1$ and $\alpha(n-1)$ is changed at stage t_1, then the following two cases hold.*

(i) If $\gamma_{k,s_1} + 2^{-h(n)-j-1} < [\![\gamma_{k,s_1}]\!]_{h(n)} + 2^{-h(n)}$, then $\gamma_{k,s_2} - \gamma_{k,s_1} > 2^{-h(n)-j-1}$.

(ii) If there is another stage $s_0 < s_1$ such that $\gamma_{k,s_0} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,s_0}]\!]_{h(n)} + 2^{-h(n)}$ and $E_{e,s_0+1} = E_{e,s_0} \cup G(e, s_0, h(n) + j + 1)$ then $\gamma_{k,s_2} - \gamma_{k,s_0} > 2^{-h(n)} \geq 2^{j+1} \cdot (\mu(E_{e,s_0+1}) - \mu(E_{e,s_0}))$.

Proof. By Lemma 3.3.2 we have $\gamma_{k,s_2} \geq [\![\gamma_{k,s_1}]\!]_{h(n)} + 2^{-h(n)}$.

In case (i), directly we get $\gamma_{k,s_2} > \gamma_{k,s_1} + 2^{-h(n)-j-1}$.

In case (ii), by Lemma 3.4.2, $\mu(E_{e,s_0+1}) - \mu(E_{e,s_0}) \leq 2^{-h(n)-j-1}$ and $\gamma_{k,s_1} \geq [\![\gamma_{k,s_0}]\!]_{h(n)} + 2^{-h(n)}$. Thus,

$$\gamma_{k,s_2} \geq \left[\!\left[[\![\gamma_{k,s_0}]\!]_{h(n)} + 2^{-h(n)}\right]\!\right]_{h(n)} + 2^{-h(n)} = [\![\gamma_{k,s_0}]\!]_{h(n)} + 2^{-h(n)} + 2^{-h(n)} > \gamma_{k,s_0} + 2^{-h(n)}.$$

Then $\gamma_{k,s_2} - \gamma_{k,s_0} > 2^{-h(n)} \geq 2^{j+1} \cdot (\mu(E_{e,s_0+1}) - \mu(E_{e,s_0}))$. □

Our strategy is then that, whenever $\mathcal{Q}_e$ requires attention, before adding any weight to α to change $\alpha(n-1)$, we check whether it is worth to do that, i.e. we only do that in two cases:

- $\gamma_{k,s_1} + 2^{-h(n)-j-1} < [\![\gamma_{k,s_1}]\!]_{h(n)} + 2^{-h(n)}$, or
- the last time when $\mathcal{Q}_e$ requires attention, E_e is changed.

We assign an appropriate interval I_e for each requirement $\mathcal{Q}_e$ according to Lemma 3.4.3, so that if during the construction the measure of E_e exceeds 2^{-j} or α completes an I_e-load process, γ_k would be forced to be larger than 1, which then contradicts our assumption that $\gamma_k \leq 1$. As before, we will make sure all I_e are disjoint with each other. Then by the definition of I_e-load process and Lemma 3.2.2, the actions for each requirement will work independently without interfering each other.

Proof of Theorem 3.4.1. Let $m_0 = 0$. For $e \geq 0$, let m_{e+1} be the least integer such that

$$\sum_{m_e < i \leq m_{e+1}} 2^{-g(i)} \geq 2^{2+j+m_e}.$$

As $\sum_{i \in \mathbb{N}} 2^{-g(i)} = \infty$, such integer can always be found. Let $q_e = 2^{m_{e+1}-m_e} - 1$.

Construction:

Let $e = \langle k, j \rangle$.

Let $\alpha_0 = 0$ and $E_{e,0} = \emptyset, r_{e,0} = 0$ for all e. During the construction, for any variable if no new value is specified then its value remains the same as in previous stage.

At stage $s+1$: Find the least number $e \leq s$ such that $\mathcal{Q}_e$ requires attention and $r_{e,s} < 2q_e$. If exists, and supposing by adding $2^{-m_{e+1}}$ to α_s the least changed bit of α within $[m_e, m_{e+1})$ will be $\alpha(n-1)$, we check and do the following.

- If $r_{e,s}$ is even and $\gamma_{k,s} + 2^{-h(n)-j-1} < [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}$, or $r_{e,s}$ is odd, we say it is an e-loading stage. Then add $2^{-m_{e+1}}$ to α and let $r_{e,s+1}$ be the least even number larger than $r_{e,s}$.

- If $r_{e,s}$ is even and $\gamma_{k,s} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,s}]\!]_{h(n)} + 2^{-h(n)}$, we say it is an e-testing stage. Then let $E_{e,s+1} = E_{e,s} \cup G(e, s, h(n) + j + 1)$ and $r_{e,s+1} = r_{e,s} + 1$.

Otherwise go to the next stage.

Verification:

From the construction we have the following trivial observations.

Observation 3.4.3.1. If s is an e-loading stage then $r_{e,s}$ is even and if s is an e-testing stage then $r_{e,s}$ is odd.

Observation 3.4.3.2. Between every two e-testing stages there is at least one e-loading stage.

Observation 3.4.3.3. For every e, there are at most q_e many e-loading stages and at most q_e many e-testing stages.

Let $e = \langle k, j \rangle$. Our verification is done by the following lemmas.

Lemma 3.4.4. *$\alpha \leq 1$ and it is a left-c.e. real.*

Proof. For every e, by Observation 3.4.3.3, $2^{-m_{e+1}}$ is added to α for at most q_e many times. Thus,

$$\alpha \leq \sum_{e\in\mathbb{N}} q_e \cdot 2^{-m_{e+1}} = \sum_{e\in\mathbb{N}} (2^{-m_e} - 2^{-m_{e+1}}) \leq 2^0 = 1,$$

and α is a left-c.e. real. □

Lemma 3.4.5. *For each e, $\mu(E_e) \leq 2^{-j}$.*

Proof. For a contradiction, suppose $\mu(E_e) > 2^{-j}$.

As there are at most q_e many e-testing stages, let $s_1 < s_2 < \cdots < s_{f+1}$ be all the e-testing stages. Let $s_0 = 0$. Then $\mu(E_{e,s_{f+1}}) > 2^{-j}$ and $\mu(E_{e,s_0}) = 0$. As E_e only changes at e-testing stages, then $E_{e,s_i-1} = E_{e,s_{i-1}}$ for all $1 \leq i \leq f+1$.

By Lemma 3.4.2, $\mu(E_{e,s_{f+1}}) - \mu(E_{e,s_{f+1}-1}) \leq 2^{-h(n)-j-1}$. While $h(n) \geq n \geq 0$, then $\mu(E_{e,s_f}) = \mu(E_{e,s_{f+1}-1}) \geq \mu(E_{e,s_{f+1}}) - 2^{-h(n)-j-1} > 2^{-j-1}$.

For all $1 \leq i \leq f$, by Observation 3.4.3.2, let t_i be an e-loading stage within stages (s_i, s_{i+1}), respectively. Then by Lemma 3.4.3,

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{j+1} \cdot (\mu(E_{e,s_i}) - \mu(E_{e,s_i-1})) = 2^{j+1} \cdot (\mu(E_{e,s_i}) - \mu(E_{e,s_{i-1}})).$$

Thus,

$$\gamma_{k,s_{f+1}-1} > \gamma_{k,s_1-1} + 2^{j+1} \cdot \sum_{1\leq k\leq f} (\mu(E_{e,s_i}) - \mu(E_{e,s_{i-1}})) \geq 2^{j+1} \cdot \mu(E_{e,s_f}) > 1.$$

Then $\gamma_k \geq \gamma_{k,s_{f+1}-1} > 1$, which contradicts our assumption that $\gamma_k \leq 1$. □

Lemma 3.4.6. *For every e, if at some stage s, $r_{e,s} = 2q_e$ then $\mathcal{Q}_e$ will never require attention at any later stage.*

Proof. For a contradiction, suppose for some e, s we have $r_{e,s} = 2q_e$ and $\mathcal{Q}_e$ requires attention at some stage $s' > s$. Then there are already q_e many e-loading stages before stage s'. Let $t_1 < t_2 < \cdots < t_f$ be these e-loading stages, where $f = q_e$. Let $t_0 = 0$. For each $1 \leq i \leq f$, if there is an e-testing stage t' within stages (t_{i-1}, t_i), let $s_i = t'$; otherwise, let $s_i = t_i$. Let $s_{f+1} = s'$. Then for any $1 \leq i \leq f$ it holds that $s_i \leq t_i < s_{i+1}$.

For some $1 \leq i \leq f$, suppose at stage t_i the least changed bit of α within $[m_e, m_{e+1})$ is $\alpha(n-1)$. Note that $\mathcal{Q}_e$ requires attention at stages s_i, t_i, s_{i+1} respectively (s_i and t_i may be identical). If s_i is an e-loading stage, $s_i = t_i$, then by case (i) of Lemma 3.4.3, we have

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{-h(n)-j-1}.$$

If s_i is an e-testing stage, $s_i < t_i$, then by case (ii) of Lemma 3.4.3, we have

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{-h(n)} > 2^{-h(n)-j-1}.$$

On the other hand, $2^{-m_{e+1}}$ is added to α at all e-loading stages. It is easy to check that $(0, t_f]$ is an α $[m_e, m_{e+1})$-load process with loading stages $\{t_i\}_{1\leq i\leq f}$. By Lemma 3.2.2 for each $n \in [m_e, m_{e+1})$, there are 2^{n-m_e} many loading stages where the least changed bit of α is $\alpha(n)$. Thus,

$$\gamma_{k,t_{f+1}-1} - \gamma_{k,t_1-1} > \sum_{m_e \leq n-1 < m_{e+1}} 2^{n-1-m_e} \cdot 2^{-h(n)-j-1} = 2^{-m_e-j-2} \cdot \sum_{m_e < n \leq m_{e+1}} 2^{-g(n)} \geq 1.$$

Then $\gamma_k \geq \gamma_{k,t_{f+1}-1} > 1$, which contradicts our assumption that $\gamma_k \leq 1$. □

Lemma 3.4.7. *For every e, requirement $\mathcal{Q}_e$ requires attention only finitely many times.*

Proof. By Lemma 3.4.6, actually $r_{e,s} < 2q_e$ already holds when $\mathcal{Q}_e$ requires attention at stage $s+1$. Thus, by construction $\mathcal{Q}_e$ may require attention only at i-loading or i-testing stages for $i \leq e$. As for every i there are only finitely many i-loading stages and i-testing stages, then $\mathcal{Q}_e$ requires attention only finitely many times. □

Lemma 3.4.7 implies that every requirement $\mathcal{Q}_e$ requires no attention from some stage on, which means that it is satisfied eventually. Thus, every requirement $\mathcal{R}_k$ is satisfied as well. This completes the proof of Theorem 3.4.1. □

We can also modify the proof of Theorem 3.4.1 to prove that actually such left-c.e. real can be constructed in every array non-computable degree.

Theorem 3.4.8 (Fang and Merkle [25])**.** *Let g be a computable slow order or constant function and* **d** *be an array non-computable c.e. degree. There is a left-c.e. real in* **d** *such that no left-c.e. Martin-Löf random real computes it with redundancy g.*

Given g as stated, let $h(n) = n+g(n)$. Let $\mathbf{d}$ be an array non-computable c.e. degree. We construct a left-c.e. real as in the proof of Theorem 3.4.1 with the extra property that it is coded by some c.e. set in $\mathbf{d}$ and also codes that set. This is also achieved by the multiple permitting argument.

At first, suppose we have already fixed some very strong array $\{F_n\}_{n\in\omega}$. Requirements $\{\mathcal{Q}_{\langle k,j\rangle}\}_{\langle k,j\rangle\in\omega}$ are defined the same as (3.9). Then we make infinitely many copies of $\mathcal{Q}_{\langle k,j\rangle}$ which are now represented by requirements $\mathcal{T}_{\langle k,j,l\rangle}$ defined as follows for all $l \in \mathbb{N}$.

$$\mathcal{T}_{\langle k,j,l\rangle} : V_{\langle k,j\rangle} \cap F_l = D \cap F_l \Rightarrow \mathcal{Q}_{\langle k,j\rangle} \text{ is satisfied,}$$

where the set $V_{\langle k,j\rangle}$ is a c.e. set we will define during the construction to handle the permission requests from $\mathcal{Q}_{\langle k,j\rangle}$.

As before, we define a state of being active or inactive for each requirement $\mathcal{T}_{\langle k,j,l\rangle}$ during the construction. At stage $s+1$, requirement $\mathcal{T}_{\langle k,j,l\rangle}$ *requires attention* if

$$l \geq \langle k,j\rangle,\ \mathcal{T}_{\langle k,j,l\rangle} \text{ is active, } V_{\langle k,j\rangle,s} \cap F_l = D_s \cap F_l \text{ and } \mathcal{Q}_{\langle k,j\rangle} \text{ requires attention.}$$

Then by replacing $\mathcal{Q}_e$ with $\mathcal{T}_e$, the construction will be almost the same as in the proof of Theorem 3.4.1, except that before adding weight to α, we always request a permission as in the proof of Theorem 3.3.7.

Proof of Theorem 3.4.8. Let $m_0 = 0$. For $e \geq 0$, let m_{e+1} be the least integer such that

$$\sum_{m_e+1<i\leq m_{e+1}} 2^{-g(i)} \geq 2^{3+j+m_e}.$$

As $\sum_{i\in\mathbb{N}} 2^{-g(i)} = \infty$, such integer can always be found. Let $I_e = [m_e+1, m_{e+1})$ and $J = \{m_e | e \in \mathbb{N}\}$. Note that J and all I_e are disjoint and their union is $\mathbb{N}$. Let $q_e = 2^{m_{e+1}-m_e-1} - 1$.

We fix a very strong array $\{F_l\}_{l\in N}$ such that

$$|F_l| = 1 + |F_{l-1}| + \max\{q_{\langle k,j,l\rangle} : \langle k,j\rangle \leq l\}.$$

By Proposition 2.3.2, let $D \in \mathbf{d}$ be a c.e. set such that

$$(\forall e)(\exists^\infty n)[W_e \cap F_n = D \cap F_n] \tag{3.14}$$

Without loss of generality, we assume $D_0 = \emptyset$.

Construction:

Let $e = \langle k, j, l\rangle$. Set all requirements $\mathcal{T}_e$ to be active.

Let $\alpha_0 = 0$ and $E_{e,0} = \emptyset, r_{e,0} = 0, V_{\langle k,j\rangle,0} = \emptyset$ for all e. During the construction, for any variable if no new value is specified then its value remains the same as in previous stage.

At stage $2s+1$*:* For each $i \in D_{s+1} \setminus D_s$, let $\alpha(m_i) = 1$.

Find the least number $e \leq 2s$ such that $\mathcal{T}_e$ requires attention and $r_{e,2s} < 2q_e$. If exists, and supposing by adding $2^{-m_{e+1}}$ to α the least changed bit of $\alpha \upharpoonright I_e$ will be $\alpha(n-1)$, we check and do the following.

- If $r_{e,2s}$ is even and $\gamma_{k,2s} + 2^{-h(n)-j-1} < [\![\gamma_{k,2s}]\!]_{h(n)} + 2^{-h(n)}$, or $r_{e,2s}$ is odd we say it is an e-pending stage and set $\mathcal{T}_e$ to be inactive. Then we pick some $x \in F_l \setminus V_{\langle k,j\rangle,2s}$ and let $V_{\langle k,j\rangle,2s+1} = V_{\langle k,j\rangle,2s} \cup \{x\}$.

- If $r_{e,2s}$ is even and $\gamma_{k,2s} + 2^{-h(n)-j-1} \geq [\![\gamma_{k,2s}]\!]_{h(n)} + 2^{-h(n)}$, we say it is an e-testing stage. Then let $E_{e,2s+1} = E_{e,2s} \cup G(e, 2s, h(n) + j + 1)$ and $r_{e,2s+1} = r_{e,2s} + 1$.

Otherwise go to the next stage directly.

At stage $2s+2$*:* Find the least number $e \leq 2s+1$ such that requirement $\mathcal{T}_e$ is inactive and $V_{\langle k,j\rangle,2s+1} \cap F_l = D_{2s+1} \cap F_l$. If exists, we say it is an e-loading stage and set $\mathcal{T}_e$ to be active. Then add $2^{-m_{e+1}}$ to α and let $r_{e,2s+2}$ be the least even number larger than $r_{e,2s+1}$. Otherwise go to the next stage directly.

Verification:

From the construction we have the following trivial observation.

Observation 3.4.8.1. If s is an e-loading stage then $r_{e,s}$ is even and if s is an e-testing stage then $r_{e,s}$ is odd.

Observation 3.4.8.2. Between every two e-testing stages s_0, s_1 there must be two stages $t_0 < t_1$ such that t_0 is an e-pending stage and t_1 is an e-loading stage.

Observation 3.4.8.3. For every e, there are at most q_e many e-testing stages, e-pending stages and e-loading stages, respectively.

Let $e = \langle k, j, l\rangle$. Our verification is done by the following lemmas.

Lemma 3.4.9. *For each $e = \langle k, j, l\rangle$, for all s, $|F_l \cap V_{\langle k,j\rangle,s}| < |F_l|$.*

Proof. First, we notice that $|F_l \cap V_{\langle k,j\rangle}|$ increases only at e-pending stages, and at each such stage it increases by 1.

If $l < \langle k, j\rangle$, $\mathcal{T}_e$ never requires attention and there is no e-pending stage. Then for all s, $|F_l \cap V_{\langle k,j\rangle,s}| = 0 < |F_l|$.

If $l \geq \langle k, j\rangle$, by definition we have $|F_l| \geq 1 + q_e$. Then by Observation 3.4.8.3 for all s, $|F_l \cap V_{\langle k,j\rangle,s}| \leq q_e < |F_l|$. □

Lemma 3.4.9 ensures that at every e-pending stage s, $F_l \setminus V_{\langle k,j\rangle,s-1} \neq \emptyset$. Thus, the permission requests are always handled.

Lemma 3.4.10. $\alpha \leq 1$ *and it is a left-c.e. real.*

Proof. For every e, by Observation 3.4.8.3, $2^{-m_{e+1}}$ is added to α for at most q_e many times. On the other hand, for each e, 2^{-m_e-1} is added to α when e enters D. While as D is a c.e. set, this happens at most once for each e. Thus,

$$\alpha \leq \sum_{e\in\mathbb{N}} q_e \cdot 2^{-m_{e+1}} + \sum_{e\in\mathbb{N}} 2^{-m_e-1} = \sum_{e\in\mathbb{N}} (2^{-m_e-1} - 2^{-m_{e+1}} + 2^{-m_e-1}) \leq 2^0 = 1,$$

and α is a left-c.e. real. □

Lemma 3.4.11. $\alpha =_T D$.

Proof. For each e, as $2^{-m_{e+1}}$ is added to α for at most q_e many times, the actions for requirement $\mathcal{T}_e$ will not affect α outside the interval I_e. Then for any $n \in \mathbb{N}$, $D(n) = 1$ if and only if $\alpha(m_n) = 1$. So $D \leq_T \alpha$.

Assume now we are given D, fix some n. If $n = m_p$ for some number p, then clearly $\alpha(n) = D(p)$. Otherwise, there must be some e such that $n \in I_e$. We find a stage s where $D_s \cap F_l = D \cap F_l$, then claim that $\alpha(n) = \alpha_{s_0}(n)$, where $s_0 = s + 2 + \sum_{i<e} 2q_i$.

Suppose not, then there is a stage $s' > s_0$ such that $\alpha_{s_0}(n) \neq \alpha_{s'}(n)$. That is to say, there should be at least one e-loading stage within stages $(s_0, s']$. Let $s_1 + 1$ be the least e-loading stage within stages $(s_0, s']$. Then $V_{\langle k,j\rangle,s_1} \cap F_l = D_{s_1} \cap F_l$.

If there is an e-pending stage $s_2 + 1$ within $(s, s_1]$, then

$$D_s \cap F_l \quad \subseteq \quad D_{s_2} \cap F_l \quad = \quad V_{\langle k,j\rangle,s_2} \cap F_l$$

$$\subsetneq \quad V_{\langle k,j\rangle,s_2+1}\cap F_l \quad = \quad V_{\langle k,j\rangle,s_1}\cap F_l \quad = \quad D_{s_1}\cap F_l \quad \subseteq \quad D\cap F_l,$$

which contradicts $D_s \cap F_l = D \cap F_l$.

If there is no e-pending stage within $(s, s_1]$, then $\mathcal{T}_e$ is inactive at all stages within $(s, s_1]$ and $V_{\langle k,j\rangle,t} \cap F_l$ does not change for all $t \in (s, s_1]$. On the other hand, as $D_s \cap F_l = D \cap F_l$, $D_t \cap F_l$ also does not change for all $t \in (s, s_1]$. Thus, for all $t \in (s, s_1]$

$$V_{\langle k,j\rangle,t}\cap F_l \quad = \quad V_{\langle k,j\rangle,s_1}\cap F_l \quad = \quad D_{s_1}\cap F_l \quad = \quad D_t\cap F_l.$$

This implies that for all $t \in (s, s_1]$, if t is even then stage t should be an i-loading stage for some $i < e$. As for each i there are at most q_i many i-loading stages, then $s_1 + 1 \leq s + 2 + \sum_{i<e} 2 \cdot q_i = s_0$, which contradicts the assumption $s_1 + 1 > s_0$. □

Lemma 3.4.12. *For each e, $\mu(E_e) \leq 2^{-j}$.*

Proof. For a contradiction, suppose $\mu(E_e) > 2^{-j}$.

As there are at most q_e many e-testing stages, let $s_1 < s_2 < \cdots < s_{f+1}$ be all the e-testing stages. Let $s_0 = 0$. Then $\mu(E_{e,s_{f+1}}) > 2^{-j}$ and $\mu(E_{e,s_0}) = 0$. As E_e only changes at e-testing stages, then $E_{e,s_i-1} = E_{e,s_{i-1}}$ for all $1 \leq i \leq f+1$.

By Lemma 3.4.2, $\mu(E_{e,s_{f+1}}) - \mu(E_{e,s_{f+1}-1}) \leq 2^{-h(n)-j-1}$. While $h(n) \geq n \geq 0$, then $\mu(E_{e,s_f}) = \mu(E_{e,s_{f+1}-1}) \geq \mu(E_{e,s_{f+1}}) - 2^{-h(n)-j-1} > 2^{-j-1}$.

For all $1 \leq i \leq f$, by Observation 3.4.8.2, let $t_i < t'_i$ be an e-pending stage and an e-loading within stages (s_i, s_{i+1}). Then by Lemma 3.4.3,

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{j+1}\cdot(\mu(E_{e,s_i}) - \mu(E_{e,s_i-1})) = 2^{j+1}\cdot(\mu(E_{e,s_i}) - \mu(E_{e,s_{i-1}})).$$

Thus,

$$\gamma_{k,s_{f+1}-1} > \gamma_{k,s_1-1} + 2^{j+1}\cdot\sum_{1\leq k\leq f}(\mu(E_{e,s_i}) - \mu(E_{e,s_{i-1}})) \geq 2^{j+1}\cdot\mu(E_{e,s_f}) > 1.$$

Then $\gamma_k \geq \gamma_{k,s_{f+1}-1} > 1$, which contradicts our assumption that $\gamma_k \leq 1$. □

Lemma 3.4.13. *For every e, if at some stage s, $r_{e,s} = 2q_e$ then $\mathcal{T}_e$ will never require attention at any later stage.*

Proof. For a contradiction, suppose for some e, s we have $r_{e,s} = 2q_e$ and $\mathcal{T}_e$ requires attention at some stage $s' > s$. Then there are already q_e many e-pending and e-loading stages before stage s'. Let $t'_1 < t'_2 < \cdots < t'_f$ be these e-pending stages and $t_1 < t_2 < \cdots < t_f$ be these e-loading stages, where $f = q_e$. Let $t_0 = 0$. For each $1 \leq i \leq f$, if there is an e-testing stage t' within stages (t_{i-1}, t'_i), let $s_i = t'$; otherwise, let $s_i = t'_i$. Let $s_{f+1} = s'$. Then for any $1 \leq i \leq f$ it holds that $s_i \leq t'_i < t_i < s_{i+1}$.

For some $1 \leq i \leq f$, suppose at stage t_i the least changed bit of $\alpha \upharpoonright I_e$ is $\alpha(n-1)$. Note that $\mathcal{Q}_{\langle k,j\rangle}$ requires attention at stages s_i, t'_i, s_{i+1} respectively (s_i and t'_i may be identical). If s_i is an e-loading stage, $s_i = t'_i$, then by case (i) of Lemma 3.4.3, we have

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{-h(n)-j-1}.$$

If s_i is an e-testing stage, $s_i < t'_i$, then by case (ii) of Lemma 3.4.3, we have

$$\gamma_{k,s_{i+1}-1} - \gamma_{k,s_i-1} > 2^{-h(n)} > 2^{-h(n)-j-1}.$$

On the other hand, $2^{-m_{e+1}}$ is added to α at all e-loading stages. It is easy to check that $(0, t_f]$ is an α I_e-load process with loading stages $\{t_i\}_{1\leq i\leq f}$. By Lemma 3.2.2 for each $n \in I_e$, there are 2^{n-m_e-1} many loading stages where the least changed bit of α is $\alpha(n)$. Thus,

$$\gamma_{k,t_{f+1}-1} - \gamma_{k,t_1-1} > \sum_{n-1\in I_e} 2^{n-1-m_e-1} \cdot 2^{-h(n)-j-1} = 2^{-m_e-j-3} \cdot \sum_{m_e+1<n\leq m_{e+1}} 2^{-g(n)} \geq 1.$$

Then $\gamma_k \geq \gamma_{k,t_{f+1}-1} > 1$, which contradicts the assumption that $\gamma_k \leq 1$. □

Lemma 3.4.14. *For every e, requirement $\mathcal{T}_e$ requires attention only finitely many times.*

Proof. By Lemma 3.4.13, actually $r_{e,2s} < 2q_e$ already holds when $\mathcal{T}_e$ requires attention at stage $2s+1$. Thus, by construction for odd stages, $\mathcal{T}_e$ may require attention only at i-pending stages for $i \leq e$. As for every i there are only finitely many i-pending stages, then $\mathcal{T}_e$ requires attention only at finitely many odd stages, and after the last such stage it will not require attention any more. □

Now fix some $\langle k, j\rangle$. As $V_{\langle k,j\rangle}$ is a c.e. set, by (3.14), there are infinitely many l such that $V_{\langle k,j\rangle} \cap F_l = D \cap F_l$. Fix some $l \geq \langle k, j\rangle$ such that $V_{\langle k,j\rangle} \cap F_l =$

$D \cap F_l$. Then there is a stage s_0 such that $V_{\langle k,j\rangle,s} \cap F_l = D_s \cap F_l$ for all $s \geq s_0$. As there are only finitely many $\langle k,j,l\rangle$-pending stages, there is a stage $s_1 \geq s_0$ such that there is no $\langle k,j,l\rangle$-pending stage after stage s_1. Then if $\mathcal{T}_{\langle k,j,l\rangle}$ is active at some stage $s' \geq s_1$, it will remain active thereafter, because it could only become inactive at $\langle k,j,l\rangle$-pending stages. If $\mathcal{T}_{\langle k,j,l\rangle}$ is inactive at stage s_1, by the same argument as in the proof of Lemme 3.4.11, $\mathcal{T}_{\langle k,j,l\rangle}$ will become active and remain active thereafter at the latest at stage $s_1 + 2 + \sum_{i<\langle k,j,l\rangle} 2q_i$. Moreover, by Lemma 3.4.14 there is a stage $s_2 \geq s_1 + 2 + \sum_{i<\langle k,j,l\rangle} 2q_i$ such that for all $s \geq s_2$, requirement $\mathcal{T}_{\langle k,j,l\rangle}$ does not require attention, which then implies $\mathcal{Q}_{\langle k,j\rangle}$ does not require attention. Thus, $\mathcal{Q}_{\langle k,j\rangle}$ is satisfied eventually. Therefore every requirement $\mathcal{R}_k$ is satisfied. This completes the proof of Theorem 3.4.8.

□

3.5 A.N.C. Degrees and Coding with Small Redundancy

By Theorems 3.3.7, 3.4.8 and 2.3.3, it is easy to get the following extension of Theorem 2.3.3.

Theorem 3.5.1 (Fang and Merkle [25])**.** *Let g be a computable slow order or a constant function. The following are equivalent for a c.e. degree* $\mathbf{d}$*:*

1. *There are two left-c.e. reals in* $\mathbf{d}$ *such that no left-c.e. real codes both of them with redundancy g.*

2. *There is a left-c.e. real in* $\mathbf{d}$ *such that no left-c.e. random real codes it with redundancy g.*

3. *There is a set in* $\mathbf{d}$ *such that no left-c.e. random real codes it with redundancy g.*

4. $\mathbf{d}$ *is array non-computable.*

Thus, for coding with small redundancy, the extended Yu-Ding Theorem 3.3.1 and the extended Barmpalias-Lewis Theorem 3.4.1 are still characterizations of array non-computable degrees.

3.6 Summary

In this chapter we showed some examples where theorems about cl-reducibility, which is coding with constant redundancy, can be extended to coding with low redundancy. Basically, we extended the Yu-Ding Theorem and the Barmpalias-Lewis Theorem from coding with constant redundancy to coding with small redundancy. One the other hand, we also showed that such extensions are valid only for small redundancy. Moreover, in the same way as showed, the result that the Yu-Ding Theorem and the Barmpalias-Lewis Theorem are characterizations of c.e. array non-computable degrees, still holds for the extended versions of both theorems.

However, there are also examples where such kind of extension fail. Fan and Yu [24] showed that there is a left-c.e. real not cl-computable from any left-c.e. complex real. And Ambos-Spies, Losert, and Monath [3] showed that such left-c.e. reals exist in every not totally ω-c.e. degrees. We cannot extent either of these results to computations with any unbounded redundancy.

By theorem 3.1.1, with large coding redundancy, all left-c.e. random reals have the same coding power within all left-c.e. reals. But in general it is not clear under small coding redundancy, is there necessarily a difference among left-c.e. random reals in terms of coding power.

As reported in [10, Section 6], Frank Stephan proved that there are two left-c.e. random reals that one is not coded by the other with constant redundancy. Barmpalias and Lewis-Pye [7] extended the redundancy bound to $\lfloor \log n \rfloor$. Although $\lfloor \log n \rfloor$ is quite close to low redundancy in general, the following question about coding with low redundancy is still open.

Question 1. *Is it true that for any computable slow order g, there exists two left-c.e. random reals such that one is not coded by the other with redundancy g?*

Chapter 4

Martingales as Betting Strategies

In § 2.1, we introduced the notion of Martin-Löf randomness. It is characterized both by Martin-Löf tests and Solovay tests, which reflect the typicalness of a random sequences, and also characterized by prefix-free Kolmogorov complexity, which reflects the incompressibility of a random sequence. Apart from these two approaches to randomness, there is a third one which characterizes its unpredictability. Intuitively, in a random process such as repeatedly tossing a fair coin, results of all the previous tosses should be of no help for predicting the result of the next toss. Then if we are betting on the successive bits of a random sequence, we should not expect to be able to make much money, no matter what betting strategy we apply. Usually, in a binary betting game, a betting strategy determines to which outcome it bets and how much capital it puts on it, or to what percentage of capital it puts on it. This process can be presented by a martingale which records the capital after each betting stage. If along some sequence of bits, the capital recorded by the martingale is unbounded, we would say such a sequence is not random. In this way, we define a sequence to be random if there is no martingale reaches unbound value along it. Of course, to realize this idea we need to specify the effectiveness of the martingales. Such candidates could be computable martingales or left-c.e. martingales. Interestingly, as a basic result in algorithmic randomness, in the case of left-c.e. martingales, the resulting randomness notion coincides with Martin-Löf randomness.

In this chapter we introduce the standard notion of martingales in detail and develop some new notions. In the meanwhile we also explore some of the relevant properties of these notions to give a better understanding of them and for later references as well.

4.1 Martingales and Supermartingales

Let us begin with the formal definition of martingales.

Definition 4.1.1. A function $M\colon 2^{<\omega} \mapsto \mathbb{R}^{0+}$ is a *martingale* if for all $\sigma \in 2^{<\omega}$,

$$M(\sigma) = \frac{M(\sigma^\frown 0) + M(\sigma^\frown 1)}{2}. \tag{4.1}$$

It is a *supermartingale* if for all $\sigma \in 2^{<\omega}$,

$$M(\sigma) \geq \frac{M(\sigma^\frown 0) + M(\sigma^\frown 1)}{2}. \tag{4.2}$$

We also define some further notions for the analysis of (super)martingales.

Definition 4.1.2. Let M be a (super)martingale.

1. The *wager* w_M of M is defined as

$$w_M(\sigma^\frown i) = \frac{M(\sigma^\frown i) - M(\sigma^\frown (1-i))}{2} \tag{4.3}$$

   for all $\sigma \in 2^{<\omega}$ and $i \in \{0,1\}$; and $w_M(\lambda) = 0$.

2. The *cover* $\widehat{M}$ of M is defined as

$$\widehat{M}(\sigma) = M(\lambda) + \sum_{\tau \preceq \sigma} w_M(\tau) \tag{4.4}$$

   for all $\sigma \in 2^{<\omega}$.

3. The *marginal saving* M^* of M is defined as

$$M^*(\sigma) = M(\sigma) - \frac{M(\sigma^\frown 0) + M(\sigma^\frown 1)}{2} \tag{4.5}$$

   for all $\sigma \in 2^{<\omega}$.

4. The *(accumulated) saving* S_M of M is defined as

$$S_M(\sigma) = \sum_{\tau \prec \sigma} M^*(\tau) \tag{4.6}$$

for all $\sigma \in 2^{<\omega}$.

For a wager function w_M, we can interpret $w_M(\sigma ^\frown i)$ as the wager M putting on outcome i at stage σ. By definition $M^*(\sigma) \geq 0$ for all $\sigma \in 2^{<\omega}$. So S_M is nondecreasing. We have some simple observations for these notations.

Proposition 4.1.3. *Given a (super)martingale M, for any $\sigma \in 2^{<\omega}$ and $i \in \{0,1\}$ the following are true.*

(i) $w_M(\sigma ^\frown i) = -w_M(\sigma ^\frown (1-i))$.

(ii) $w_M(\sigma ^\frown i) \geq M(\sigma ^\frown i) - M(\sigma)$, *the equality holds when M is a martingale.*

(iii) $|w_M(\sigma ^\frown i)| \leq M(\sigma)$.

(iv) $\widehat{M}$ *is a martingale, and* $w_M(\sigma) = w_{\widehat{M}}(\sigma)$.

(v) $S_M(\sigma) = \widehat{M}(\sigma) - M(\sigma)$.

(vi) If M is a martingale, $\widehat{M}(\sigma) = M(\sigma)$.

Proof. (i): This is obvious by definition.

(ii): By adding up (4.3) and (4.2) or (4.1) we get $w_M(\sigma ^\frown i) + M(\sigma) \geq M(\sigma ^\frown i)$, where the equality holds when M is a martingale.

(iii): As $M(\tau) \geq 0$ for all $\tau \in 2^{<\omega}$, by (ii) we have $w_M(\sigma ^\frown i) \geq -M(\sigma)$ and $w_M(\sigma ^\frown (1-i)) \geq -M(\sigma)$. Then by applying (i) we have $|w_M(\sigma ^\frown i)| \leq M(\sigma)$.

(iv): By definition and (i)

$$\frac{\widehat{M}(\sigma ^\frown 0) + \widehat{M}(\sigma ^\frown 1)}{2} = M(\lambda) + \sum_{\tau \preceq \sigma} w_M(\tau) + \frac{w_M(\sigma ^\frown 0) + w_M(\sigma ^\frown 1)}{2} = \widehat{M}(\sigma).$$

So $\widehat{M}$ is a martingale. And

$$w_{\widehat{M}}(\sigma ^\frown i) = \frac{\widehat{M}(\sigma ^\frown i) - \widehat{M}(\sigma ^\frown (1-i))}{2} = \frac{w_M(\sigma ^\frown i) - w_M(\sigma ^\frown (1-i))}{2} = w_M(\sigma ^\frown i).$$

(v): We prove by induction. At first, $S_M(\lambda) = 0 = \widehat{M}(\lambda) - M(\lambda)$. Suppose we have $S_M(\sigma) = \widehat{M}(\sigma) - M(\sigma)$. Then

$$\begin{aligned} S_M(\sigma^\frown i) &= S_M(\sigma) + M^*(\sigma) \\ &= \widehat{M}(\sigma) - M(\sigma) + M(\sigma) - \frac{M(\sigma^\frown 0) + M(\sigma^\frown 1)}{2} \\ &= \widehat{M}(\sigma^\frown i) - \frac{M(\sigma^\frown i) - M(\sigma^\frown (1-i))}{2} - \frac{M(\sigma^\frown 0) + M(\sigma^\frown 1)}{2} \\ &= \widehat{M}(\sigma^\frown i) - M(\sigma^\frown i). \end{aligned}$$

(vi): If M is a martingale, $M^*(\tau) = 0$ for all $\tau \in 2^{<\omega}$ and then S_M is constantly 0. By (v), $\widehat{M}(\sigma) - M(\sigma) = S_M(\sigma) = 0$. □

The following proposition can also be easily verified. It is useful to construct supermartingales.

Proposition 4.1.4. *Given a supermartingale M and a nondecreasing function $f\colon 2^{<\omega} \mapsto \mathbb{R}$, if $f(\sigma) \leq \widehat{M}(\sigma)$ for all $\sigma \in 2^{<\omega}$, then $\widehat{M} - f$ is a supermartingale.*

Sometimes it is convenient to have the following multiplicative form for a (super)martingale.

Definition 4.1.5. Let M be a (super)martingale. The *betting coefficient* of M is defined as

$$c_M(\sigma^\frown i) = \begin{cases} \frac{M(\sigma^\frown i)}{M(\sigma)} & \text{if } M(\sigma) > 0, \\ 1 & \text{otherwise,} \end{cases} \tag{4.7}$$

for all $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$; and $c_M(\lambda) = 1$.

For this notation, the following properties follow obviously.

Proposition 4.1.6. *Given a (super)martingale M, for any $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$ the following are true.*

1. $0 \leq c_M(\sigma) \leq 2$.
2. $c_M(\sigma^\frown 0) + c_M(\sigma^\frown 1) \leq 2$, *the equality holds when M is a martingale.*
3. $M(\sigma) = M(\lambda) \cdot \prod_{\tau \preceq \sigma} c_M(\tau)$.
4. *If M is a martingale,* $w_M(\sigma^\frown i) = (c_M(\sigma^\frown i) - 1) \cdot M(\sigma)$.

If we view a computable martingale M as a function satisfying (4.1), then it provides a formalization of a betting strategy on an infinite coin-tossing game: at position σ our capital is $M(\sigma)$ and we bet $w_M(\sigma^\frown i)$ (or $(c_M(\sigma^\frown i) - 1) \times 100\%$ of the current capital) on i for $i \in \{0,1\}$. Note that either $w_M(\sigma^\frown 0) = w_M(\sigma^\frown 1) = 0$ $(c_M(\sigma^\frown 0) = c_M(\sigma^\frown 1) = 1)$, in which case no bet is placed for the next bit of σ, or for some $i \in \{0,1\}$ we have $w_M(\sigma^\frown i) > 0, w_M(\sigma^\frown (1-i)) < 0$, $(c_M(\sigma^\frown i) > 1 > c_M(\sigma^\frown (1-i)))$ which means that the bet is put on i at this stage. In the same spirit we can turn a supermartingale M into a betting strategy, while at every stage some capital $M^*(\sigma)$ is lost, either consumed or saved into a frozen account. In this sense, supermartingales often model betting strategies that also incorporate consumption or savings during the game, or even betting strategies that operate under inflation.

4.2 Computable (Super)martingales and Their Mixtures

In order to consider realistic betting strategies it is natural to require that the (super)martingales to be somehow effective. The most common examples are computable martingales and left-c.e. martingales. For convenience, we denote the class of all computable martingales by CM, the class of all computable supermartingales by CS, the class of all left-c.e. martingales by LM, the class of all left-c.e. supermartingales by LS.

Inspired by the fact that left-c.e. nonnegative functions are mixtures of uniformly computable nonnegative functions, we also define another class of effective supermartingales.

Definition 4.2.1. A (super)martingale is called *strongly left-c.e.* if it is a mixture of uniformly computable (super)martingales.

We denote the class of all strongly left-c.e. martingales by SLM, and the class of all strongly left-c.e. supermartingales by SLS. Table 4.1 summarizes our notions of classes of effective betting strategies.

Clearly, all strongly left-c.e. (super)martingales are left-c.e. (super)martingales. In general, we don't know whether the converse is true, whereas the following proposition shows that for martingales it is true.

CM	the class of all computable martingales
CS	the class of all computable supermartingales
LM	the class of all left-c.e. martingales
LS	the class of all left-c.e. supermartingales
SLM	the class of all strongly left-c.e. martingales
SLS	the class of all strongly left-c.e. supermartingales

Table 4.1 Classes of effective betting strategies

Proposition 4.2.2. *A martingale is left-c.e. if and only if it is a mixture of uniformly computable martingales.*

Proof. If $\{M_i\}$ is a family of uniformly computable martingales, $\sum_i M_i(\lambda) < \infty$ and $M(\sigma) = \sum_i M_i(\sigma)$, then it is easy to see that M is a left-c.e. martingale.

For the converse, let M be a left-c.e. martingale. We assume M is not a computable martingale, because otherwise it is trivial. Then there exists a left-c.e. approximation $\{M_s\}$ to M such that $M_{s+1}(\sigma) > M_s(\sigma)$ for all s, σ. We define a family $\{N_i\}_{i\in\omega}$ of uniformly computable martingales as follows. Inductively assume that $\{N_i\}_{i<k}$ have been defined, they are computable martingales, and

$$S_k(\sigma) < M(\sigma) \quad \text{for all } \sigma, \text{ where } S_k := \sum_{i<k} N_i. \tag{4.8}$$

Then there is a stage s_0 such that $M_{s_0}(\lambda) > S_k(\lambda)$. Let $N_k(\lambda) = M_{s_0}(\lambda) - S_k(\lambda)$. For each σ suppose inductively that we have defined $N_k(\sigma)$ in such a way that $N_k(\sigma) + S_k(\sigma) \leq M_t(\sigma)$ for some stage $t > s_0$. Since M is a martingale, this means that there exists some larger stage s such that:

$$M_s(\sigma^\frown 0) + M_s(\sigma^\frown 1) \geq 2N_k(\sigma) + 2S_k(\sigma) = 2N_k(\sigma) + (S_k(\sigma^\frown 0) + S_k(\sigma^\frown 1)).$$

Then we let $N_k(\sigma^\frown i), i = \{0, 1\}$ be two non-negative rationals such that:

1. $N_k(\sigma^\frown 0) + N_k(\sigma^\frown 1) = 2N_k(\sigma)$;
2. $N_k(\sigma^\frown i) + S_k(\sigma^\frown i) \leq M_s(\sigma^\frown i)$ for each $i = \{0, 1\}$.

This concludes the inductive definition of N_k and also verifies the property (4.8) for $k + 1$ in place of k. Note that the totality of each N_i is guaranteed by the fact that M is a martingale. It remains to show that

$$\lim_k S_k(\sigma) = M(\sigma) \quad \text{for each } \sigma. \tag{4.9}$$

By the definition of $N_i(\lambda)$, it follows that (4.9) holds for $\sigma = \lambda$. Assuming (4.9) for σ, we show that it holds for $\sigma \frown i, i \in \{0,1\}$. We have

$$M(\sigma \frown 0)+M(\sigma \frown 1)-S_k(\sigma \frown 0)-S_k(\sigma \frown 1) = 2M(\sigma)-2S_k(\sigma) = 2(M(\sigma)-S_k(\sigma)).$$

So by (4.9) we have $\lim_k S_k(\sigma \frown 0) + \lim_k S_k(\sigma \frown 1) = M(\sigma \frown 0) + M(\sigma \frown 1)$. By (4.8) applied to $\sigma \frown 0$ and $\sigma \frown 1$ we get $\lim_k S_k(\sigma \frown i) = M(\sigma \frown i)$ for $i \in \{0,1\}$, as required. This concludes the inductive proof of (4.9). $\square$

By Proposition 4.2.2, we have

$$\mathsf{LM} = \mathsf{SLM}. \tag{4.10}$$

Then clearly, the following holds

$$\begin{array}{ccccc} \mathsf{CM} & \subseteq & \mathsf{SLM} & = & \mathsf{LM} \\ \cap & & \cap & & \cap \\ \mathsf{CS} & \subseteq & \mathsf{SLS} & \subseteq & \mathsf{LS} \end{array} \tag{4.11}$$

For a class of (super)martingales $\mathcal{C}$, we say it is *cover closed* if for every (super)martingale M in $\mathcal{C}$ its cover $\widehat{M}$ is also in $\mathcal{C}$. Clearly, all classes of martingales are cover closed because the cover of a martingale is itself. Besides that also note that $\mathsf{CS}, \mathsf{SLS}$ are cover closed, whereas it is not clear whether LS is cover closed.

4.3 Success of (Super)martingales

Here we define three notions of success for (super)martingales.

Definition 4.3.1. Given a (super)martingale M and a sequence X, we say M

- *succeeds* on X, if $\limsup_{n\to\infty} M(X \upharpoonright n) = \infty$;
- *strongly succeeds* on X, if $\lim_{n\to\infty} M(X \upharpoonright n) = \infty$;
- *successfully saves* on X, if $\lim_{n\to\infty} S_M(X \upharpoonright n) = \infty$.

Let $\mathsf{Succ}(M)$ denote the set of all sequences on which M succeeds; $\mathsf{SSucc}(M)$ denote the set of all sequences on which M strongly succeeds; and $\mathsf{Save}(M)$ denote the set of all sequences on which M successfully saves.

And if $\mathcal{C}$ is a class of (super)martingales, we define $\mathsf{Succ}[\mathcal{C}]$ to be the collection of all sequences on which some (super)martingale in $\mathcal{C}$ succeeds, i.e.

$$\mathsf{Succ}[\mathcal{C}] = \bigcup_{M \in \mathcal{C}} \mathsf{Succ}(M).$$

$\mathsf{SSucc}[\mathcal{C}]$ and $\mathsf{Save}[\mathcal{C}]$ are defined similarly.

For two supermartingales M and N, we say that M is *superior* to N (or N is *inferior* to M) if $\mathsf{Succ}(N) \subseteq \mathsf{Succ}(M)$. A class $\mathcal{C}$ of supermartingales is *superior* to N if $\mathsf{Succ}(N) \subseteq \mathsf{Succ}[\mathcal{C}]$.

The following theorem leads to a characterization of Martin-Löf random sequences by left-c.e. martingales. More details about this aspect can be found in [21, §6.3] or the paper by Bienvenu, Shafer, and Shen [15].

Theorem 4.3.2 (Schnorr [42, 43])**.**

$$\mathrm{MLR} = 2^{\omega} \setminus \mathsf{Succ}[\mathit{LM}].$$

The proof of this theorem is based on the following result.

Theorem 4.3.3 (Kolmogorov's Inequality, Ville [51])**.** *Let M be a (super)martingale. For any string σ and any prefix-free set S of extensions of σ, we have*

$$\sum_{\tau \in S} 2^{-|\tau|} M(\tau) \leq 2^{-|\sigma|} M(\sigma).$$

Now we explore some properties about these three success notions. By definition, it is easy to get the following proposition.

Proposition 4.3.4. *For a (super)martingale M, it holds that*

$$\mathsf{SSucc}(M) \subseteq \mathsf{Succ}(M) \textit{ and } \mathsf{Save}(M) \subseteq \mathsf{SSucc}(\widehat{M}).$$

By Proposition 4.3.4 the following is also true.

Corollary 4.3.5. *For a cover closed class $\mathcal{C}$ of (super)martingales, it holds that*

$$Save[\mathcal{C}] \subseteq SSucc[\mathcal{C}] \subseteq Succ[\mathcal{C}]$$

The following proposition is usually referred as "savings trick". We include a proof here for later reference.

Proposition 4.3.6 (Folklore)**.** *Given any supermartingale M, there exists a supermartingale N computable from M such that $\mathsf{Succ}(M) \subseteq \mathsf{Save}(N)$.*

Proof. We enumerate a set S of strings as marks to mark all the places where M doubles its capital with respect to the value of M at the latest marked place. The set S is enumerated by induction on the length of strings. First enumerate λ into S. Suppose we have already enumerated all the marks of length less than n into S. For $\sigma \in 2^n$, let τ be the longest prefix of σ in S (note that such τ always exists as $\lambda \in S$). If $M(\sigma) \geq 2M(\tau)$, enumerate σ into S.

Then we construct N according to the same betting coefficient as M, while saving 1 at the all marked places. Let c_M be the betting coefficient of M. Formally, we let $N(\lambda) = 2$. For all $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$, let

$$N(\sigma^\frown i) = \begin{cases} N(\sigma) \cdot c_M(\sigma^\frown i) & \text{if } \sigma \notin S; \\ (N(\sigma) - 1) \cdot c_M(\sigma^\frown i) & \text{if } \sigma \in S. \end{cases}$$

Lemma 4.3.7. $N(\sigma) \geq 2$ *for all* $\sigma \in S$.

Proof. We prove by induction on length of strings. By definition $N(\lambda) = 2$. Suppose $N(\sigma) \geq 2$ for all $\sigma \in S \cap 2^{<n}$ and we are given $\sigma' \in S \bigcap 2^n$. Let τ be the longest prefix of σ' in S. By definition, $M(\sigma') \geq 2M(\tau)$. Thus,

$$N(\sigma') = (N(\tau) - 1) \prod_{\tau \prec \gamma \preceq \sigma'} c_M(\gamma) = (N(\tau) - 1) \frac{M(\sigma')}{M(\tau)} \geq 2(N(\tau) - 1) \geq 2.$$

$\square$

By Lemma 4.3.7, it is easy to check that $N(\sigma) \geq 0$ for all $\sigma \in 2^{<\omega}$, thus N is indeed a supermartingale.

Lemma 4.3.8. $N^*(\sigma) \geq 1$ *for all* $\sigma \in S$.

Proof. For all $\sigma \in S$, we notice that

$$N^*(\sigma) = N(\sigma) - (N(\sigma) - 1) \cdot \frac{c_M(\sigma ^\frown 0) + c_M(\sigma ^\frown 1)}{2} \geq 1.$$

$\square$

Lemma 4.3.9. *If $X \in \mathsf{Succ}(M)$, then there are infinitely many σ in S such that $\sigma \prec X$.*

Proof. We proof by contradiction. Given $X \in \mathsf{Succ}(M)$, suppose there are only finitely many σ in S such that $\sigma \prec X$. Let τ be the longest prefix of X in S. As $X \in \mathsf{Succ}(M)$, then $\limsup_{n\to\infty} M(X \upharpoonright n) = \infty$. Then there must exist some $m > |\tau|$ such that $M(X \upharpoonright n) \geq 2M(\tau)$, which means that $X \upharpoonright n$ or some of its prefix longer than τ must have been enumerated into S. This contradicts the choice of τ. $\square$

Lemma 4.3.8 and 4.3.9 together show that if $X \in \mathsf{Succ}(M)$ then $X \in \mathsf{Save}(N)$. Thus, $\mathsf{Succ}(M) \subseteq \mathsf{Save}(N)$. $\square$

From Proposition 4.3.6 we conclude that

$$\mathsf{Succ}[\mathit{CS}] \subseteq \mathsf{Save}[\mathit{CS}]. \tag{4.12}$$

However, we can not get something similar for left-c.e. supermartingales by directly applying Proposition 4.3.6, because a supermartingale computable from a left-c.e. supermartingale is not necessarily a left-c.e. martingale. However, we can still get the following proposition by an indirect construction with Martin-Löf tests, a proof of which can be found at Downey and Hirschfeldt [21, Proposition 6.3.8].

Proposition 4.3.10 (Folklore)**.** *Given any left-c.e. supermartingale M, there exists a left-c.e martingale N such that* $\mathsf{Succ}(M) \subseteq \mathsf{SSucc}(N)$.

From Proposition 4.3.10 we can conclude that

$$\mathsf{Succ}[\mathit{LS}] \subseteq \mathsf{SSucc}[\mathit{LM}]. \tag{4.13}$$

Corollary 4.3.11.

$$\mathsf{Succ}[\mathit{CM}] = \mathsf{SSucc}[\mathit{CM}] = \mathsf{Succ}[\mathit{CS}] = \mathsf{SSucc}[\mathit{CS}] = \mathsf{Save}[\mathit{CS}].$$

Proof. By Corollary 4.3.5 and (4.12), we have the following observation which concludes the above equations:

$$
\begin{tikzcd}[column sep=small]
 & \mathsf{SSucc}[\mathit{CS}] \arrow[dr, phantom, sloped, "\subseteq"] & \\
\mathsf{Save}[\mathit{CS}] \subseteq \mathsf{SSucc}[\mathit{CM}] \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \mathsf{Succ}[\mathit{CS}] \subseteq \mathsf{Save}[\mathit{CS}]. \\
 & \mathsf{Succ}[\mathit{CM}] \arrow[ur, phantom, sloped, "\subseteq"] &
\end{tikzcd}
$$

□

Corollary 4.3.12.

$$\mathsf{Succ}[\mathit{LM}] = \mathsf{SSucc}[\mathit{LM}] = \mathsf{Succ}[\mathit{SLS}] = \mathsf{SSucc}[\mathit{SLS}] = \mathsf{Succ}[\mathit{LS}] = \mathsf{SSucc}[\mathit{LS}].$$

Proof. By (4.11), (4.10), Corollary 4.3.5 and (4.13), we have the following observation which concludes the above equations:

$$
\begin{tikzcd}[column sep=small]
 & \mathsf{SSucc}[\mathit{SLS}] \subseteq \mathsf{SSucc}[\mathit{LS}] \arrow[dr, phantom, sloped, "\subseteq"] & \\
\mathsf{SSucc}[\mathit{LM}] \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \mathsf{Succ}[\mathit{LS}] \subseteq \mathsf{SSucc}[\mathit{LM}]. \\
 & \mathsf{Succ}[\mathit{LM}] \subseteq \mathsf{Succ}[\mathit{SLS}] \arrow[ur, phantom, sloped, "\subseteq"] &
\end{tikzcd}
$$

□

Note that for any martingale M, we always have $\mathsf{Save}(M) = \emptyset$. So $\mathsf{Save}[\mathit{CM}] = \mathsf{Save}[\mathit{LM}] = \emptyset$.

Given a class $\mathcal{C}$ of (super)martingales, the class of random sequences induced by $\mathcal{C}$ is $2^\omega \setminus \mathsf{Succ}[\mathcal{C}]$, and such sequences are called $\mathcal{C}$-random sequences. The above two corollaries indicate that when deal with computable or left-c.e. (super)martingales we can often use martingales or supermartingales interchangeably, and use Succ or SSucc interchangeably. Moreover, LM, SLS and LS all induce the same randomness notion as Martin-Löf randomness. This somehow justifies our definition of strongly left-c.e. supermartingales as well.

4.4 Effective Hausdorff Dimension

When Schnorr [42, 43] introduced the martingale approach to algorithmic information theory, he also showed some interest in the rate of success of (super)martingales M, and in particular the classes

$$S_h(M) = \left\{ X \colon \limsup_n \frac{M(X \upharpoonright n)}{h(n)} = \infty \right\}$$

where $h \colon \mathbb{N} \mapsto \mathbb{N}$ is a computable non-decreasing function. Later Lutz [32, 33] showed that the Hausdorff dimension of a class of sequences can be characterized by the exponential "success rates" of left-c.e. supermartingales, and in that light defined the effective Hausdorff dimension $\dim(X)$ of a sequence X as the infimum of the $s \in (0,1)$ such that $X \in S_h(M)$ for some left-c.e. supermartingale M, where $h(n) = 2^{(1-s)n}$.

Surprisingly, there are also characterizations of effective Hausdorff dimension in terms of Kolmogorov complexity and tests.

Mayordomo [36] showed that

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X \upharpoonright n)}{n} = \liminf_n \frac{\mathrm{K}(X \upharpoonright n)}{n}. \tag{4.14}$$

Given $s \in (0,1)$, an s-test is a family $\{V_i\}_{i\in\omega}$ of uniformly c.e. sets of strings such that $\sum_{\sigma\in V_i} 2^{-s|\sigma|} < 2^{-i}$ for each i. Then X is said to be weakly s-random if it avoids all s-tests $\{V_i\}_{i\in\omega}$, in the sense that there are only finitely many i such that X has a prefix in V_i. By Tadaki [48], X being weakly s-random is equivalent to the condition that $\mathrm{K}(X \upharpoonright n) > s \cdot n - \mathbf{O}(1)$ for all n. Then by (4.14) we have

$$\dim(X) = \sup\{s \mid X \text{ is weakly } s\text{-random}\}. \tag{4.15}$$

Obviously, all Martin-Löf random sequences have effective dimension 1. But it is also proved that the converse does not hold. On the other hand, there are computably random (CM-random) sequences of effective dimension 0. More on the topic of algorithmic dimension can be found in [21, Chapter 13].

Chapter 5

Betting with Preferences on Outcomes

Many real gambling systems for repeated betting are based on the strategies where they make elaborate choices for the wagers at each stage, while leaving the choice of outcome constant. Consider the game of roulette, for example, and the binary outcome of red/black.[1] Perhaps the most infamous roulette system is the "martingale",[2] where one constantly bets on a fixed color, say *red*, starts with an initial wager x and doubles the wager after each loss. At the first winning stage all losses are then recovered and an additional profit x is achieved. Such systems rely on the fairness of the game, in the form of a law of large numbers that has to be obeyed in the limit (and, of course, require unbounded initial capital in order to guarantee success with probability 1). In the example of the "martingale" the relevant law is that, with probability 1, there must be a round where the outcome is red. Many other systems have been developed that use more tame series of wagers (compared to the exponential increase of the "martingale"), and which appeal to various forms of the law of large numbers.[3]

[1] Roulettes have a third outcome 0, which is neither red nor black, and which gives a slight advantage to the house. For simplicity in our discussion we ignore this additional outcome.

[2] for the origin of this term, its use as a betting system and its adoption in mathematics, see [34] and [45].

[3] Well-known systems of this kind are: the D'Alembert System, the Fibonacci system, the Labouchère system or split martingale, and many others. See, for example, `https://www.roulettesystems.com`.

In this chapter we discuss effective betting strategies with favorable outcomes. For such strategies positive wager can only be placed on every favorable outcome. If the strategy is a mixture of betting strategies, we also require that every component has the same favorable outcomes. So we also call them *monotonous* betting strategies. One special case is when the favorable outcomes are constant, i.e. at all stages the favorable outcomes are either always 0 or always 1, in which case we say it is *single-sided*, or *0-sided* and *1-sided*, respectively. There is also a variation where it is not allowed to use earnings from the successful bets on 0s in order to bet on 1s, and vice-versa. Such a betting strategy can be presented by the sum of a 0-sided strategy and a 1-sided strategy. So we call it a *separable* strategy. We study the question that whether such a restriction will weaken the power of a class of effective betting strategies, i.e. reduce their success sequences. It turns out that the answer diverges between the class of computable betting strategies and the class of left-c.e. betting strategies.

Most of the results presented here coincide with the paper by Barmpalias, Fang, and Lewis-Pye [14], though some of the notion is slightly different.

5.1 Monotonous (Super)martingales

We now formally define strategies that bet in a monotonous fashion, in terms of (super)martingales. Recall that we denote the wager of a (super)martingale M by w_M.

Definition 5.1.1. A (super)martingale M is

- *0-sided* if $w_M(\sigma ^\frown 0) \geq 0$ for all σ;
- *1-sided* if $w_M(\sigma ^\frown 1) \geq 0$ for all σ;
- *single-sided* if it is either 0-sided or 1-sided;
- *separable* if it is the sum of a 0-sided and a 1-sided martingale

With the following notion of prediction function, we can generalize the notion of monotonous (super)martingales.

Definition 5.1.2. A *prediction function* is a (partial) function from $2^{<\omega}$ to $\{0, 1\}$. Given a prediction function f and a string σ, we say that $i < |\sigma|$ is a correct f-guess along σ if $f(\sigma \upharpoonright i) \downarrow= \sigma(i)$. The f-guess correct rate along σ is the ratio between the number of correct f-guesses along σ and the length of σ, i.e. $|\{i < |\sigma| \colon f(\sigma \upharpoonright i) \downarrow= \sigma(i)\}|/|\sigma|$.

Definition 5.1.3. Given a prediction function f, a (super)martingale M is

- *f-sided* if for all σ, $w_M(\sigma ^\frown i) > 0$ only if $f(\sigma) \downarrow= i$;
- *decidably-sided* if it is f-sided for a computable prediction function f.

Given a (super)martingale M, we define its *preference function* as follows,

$$f(\sigma) = \begin{cases} i & \text{if } (\exists i \in \{0,1\})[M(\sigma ^\frown i) > M(\sigma ^\frown (1-i))], \\ \uparrow & \text{otherwise.} \end{cases}$$

Then a supermartingale M with preference function f is an f-sided supermartingale.

A computable/left-c.e. f-sided (super)martingale is a computable/left-c.e. (super)martingale which is f-sided. We also define a strongly left-c.e. version of f-sided (super)martingales.

Definition 5.1.4. A *strongly left-c.e. f-sided* (super)martingale is a mixture of uniformly computable f-sided (super)martingales.

Definition 5.1.5. Given a left-c.e. f-sided (super)martingale M with computable approximation $\{M_s\}_{s\in\omega}$, $\{M_s\}_{s\in\omega}$ is called a *canonical approximation* if for every $s \geq 0$, $M_{s+1} - M_s$ is also an f-sided (super)martingale.

Obviously, every strongly left-c.e. f-sided (super)martingale has a canonical approximation.

It is clear that f-sided and separable (super)martingales are closed under (countable, subject to convergence of initial capitals) addition and multiplication by a constant. Thus, given a canonical approximation $\{M_i\}$ of a left-c.e. f-sided (super)martingale M, the intermediate bets $M_t - M_s$ are f-sided for any $s < t$.

We know that there is an effective list of computable approximations to all left-c.e. supermartingales with initial capital less than 1. From this list one can effectively get an effective list of canonical computable approximations to all

sCM	the class of all computable single-sided martingales
sCS	the class of all computable single-sided supermartingales
sLM	the class of all left-c.e. single-sided martingales
sLS	the class of all left-c.e. single-sided supermartingales
sSLM	the class of all strongly left-c.e. single-sided martingales
sSLS	the class of all strongly left-c.e. single-sided supermartingales
dCM	the class of all computable decidably-sided martingales
dCS	the class of all computable decidably-sided supermartingales
dLM	the class of all left-c.e. decidably-sided martingales
dLS	the class of all left-c.e. decidably-sided supermartingales
dSLM	the class of all strongly left-c.e. decidably-sided martingales
dSLS	the class of all strongly left-c.e. decidably-sided supermartingales

Table 5.1 Classes of effective monotonous betting strategies

left-c.e. supermartingales with initial capital less than 1. Note that this list includes canonical computable approximations to all strongly left-c.e. decidably-sided supermartingales with initial capital less than 1. Given a computable prediction function f, from the same list one can effectively get an effective list of all canonical computable approximations to left-c.e. f-sided supermartingales with initial capital less than 1, which includes canonical computable approximations to all strongly left-c.e. f-sided supermartingales with initial capital less than 1. Thus, there is a universal strongly left-c.e. f-sided supermartingale for every total computable prediction function f. Especially, there is a universal strongly left-c.e. 0-sided supermartingale and a universal strongly left-c.e. 1-sided supermartingale, hence also a universal strongly left-c.e. separable supermartingale.

In Table 5.1 we summarize all the classes of effective monotonous (super)martingales that we will consider in the chapter.

Many of the facts about (super)martingales in § 4.2 and §4.3 also hold for the restricted (super)martingales introduced above. Analogs of (4.11) still hold in the monotonous cases:

$$\begin{array}{ccccc} \mathsf{sCM} & \subseteq & \mathsf{sSLM} & \subseteq & \mathsf{sLM} \\ \cap & & \cap & & \cap \\ \mathsf{sCS} & \subseteq & \mathsf{sSLS} & \subseteq & \mathsf{sLS} \end{array} \quad \text{and} \quad \begin{array}{ccccc} \mathsf{dCM} & \subseteq & \mathsf{dSLM} & \subseteq & \mathsf{dLM} \\ \cap & & \cap & & \cap \\ \mathsf{dCS} & \subseteq & \mathsf{dSLS} & \subseteq & \mathsf{dLS} \end{array} . \tag{5.1}$$

A simple adaptation of the proof of Proposition 4.3.6 leads to the following proposition.

Proposition 5.1.6. *Given any f-sided supermartingale M, there exists an f-sided supermartingale N computable from M, f such that* $\mathsf{Succ}(M) \subseteq \mathsf{Save}(N)$.

Thus, we get the following analog of Corollary 4.3.11,

$$\mathsf{Succ}[\mathit{sCM}] = \mathsf{SSucc}[\mathit{sCM}] = \mathsf{Succ}[\mathit{sCS}] = \mathsf{SSucc}[\mathit{sCS}] = \mathsf{Save}[\mathit{sCS}], \tag{5.2}$$
$$\mathsf{Succ}[\mathit{dCM}] = \mathsf{SSucc}[\mathit{dCM}] = \mathsf{Succ}[\mathit{dCS}] = \mathsf{SSucc}[\mathit{dCS}] = \mathsf{Save}[\mathit{dCS}]. \tag{5.3}$$

However there is no analog of Proposition 4.3.10 for left-c.e. single-sided supermartingales, or left-c.e. decidably-sided supermartingales, because we do not have a characterization of sLS-randomness or dLS-randomness via effective tests.

Since a strongly left-c.e. single-sided (decidably-sided) supermartingale is always covered by a strongly left-c.e. single-sided (decidably-sided) martingale, sSLS and dSLS are cover closed. Then we still have

$$\mathsf{Succ}[\mathit{sSLS}] = \mathsf{Succ}[\mathit{sSLM}] \text{ and } \mathsf{Succ}[\mathit{dSLS}] = \mathsf{Succ}[\mathit{dSLM}]. \tag{5.4}$$

This means that for the strongly left-c.e. versions of monotonous betting strategies, supermartingales are interchangeable with martingales. Thus, sSLM-randomness is equivalent to sSLS-randomness, and dSLM-randomness is equivalent to dSLS-randomness.

A crucial property of a strongly left-c.e. f-sided (super)martingale S is that it is effectively approximated by f-sided computable (super)martingales $\{S_i\}$ such that for each $n < m$, *the intermediate bets $S_m - S_n$ are also f-sided*, whereas for a left-c.e. f-sided (super)martingale it is not clear whether such property also exists. One obstacle for this is the fact that *the difference of two single-sided martingales is not always single-sided*, even if they both favor the same outcome and the difference is still a martingale.

In the study of algorithmic randomness, separating different notions of randomness is often a matter of adapting existing methods on this topic, such examples can be found in [38, Chapter 7]. By adapting the arguments of [38, §7.4] directly, we can get the following theorem.

Theorem 5.1.7. *There exists X such that a 0-sided left-c.e. martingale succeeds on X and no partial computable supermartingale succeeds on X.*

Alternatively, this fact can be derived as a corollary of our Theorem 5.3.1 and the fact that the randomness notion induced by partial computable (super)martingales does not imply that the limit of the relative frequency of 0s is 1/2. Our main results in this chapter include:

$$\mathsf{Succ}[\mathit{sSLM}] \neq \mathsf{Succ}[\mathit{SLM}] \quad \text{and} \quad \mathsf{Succ}[\mathit{dSLM}] \neq \mathsf{Succ}[\mathit{SLM}],$$

which can also be viewed as results of separating randomness notions. However, the proofs require a novel argument.

5.2 Computable Single-sided Martingales

In this section we explore the succeeding power of single-sided computable martingales. First, we give two examples of types of biased sequences which can be exploited through single-sided or separable computable martingales, and we also establish basic properties of monotonous martingales that will be used later. Then we show that every computable martingale is a product of a 0-sided martingale and a 1-sided martingale, which implies that computable randomness can be characterized by computable single-sided martingales.

Success on Villes' sequence

A well-known[1] debate in the early days of probability occurred between the competing approaches of Kolmogorov, which won the debate, and the frequentist-based approach of von Mises, for the establishment of the foundations of probability. A significant factor for the loss of support to von Mises' theory was a certain sequence constructed by Ville [51][2] which is 'random' with respect to any given countable collection of choice sequences (a basic tool in von Mises'

[1]Short expositions of the debate in relation to the notion of algorithmic randomness can be found on textbooks on this topic such as the monograph by Li and Vitányi [30, §1.9] or the monograph by Downey and Hirschfeldt [21, §6.2]. Extended discussions of the philosophical underpinnings of this debate can be found in the thesis by van Lambalgen [50] and the more recent thesis by Blando [17].

[2]At `http://www.probabilityandfinance.com/misc/ville1939.pdf` an English translation can be found. Simpler proofs of Ville's theorem appear in the paper by Lieb et al. [31] and the monograph by Downey and Hirschfeldt [21, §6.5]

strictly frequentist approach) but is biased according to a well-accepted statistical test: although the frequency of 0s approaches 1/2, in all initial segments this frequency never drops below 1/2.

We point out that the bias in Villes' well-known example is exploitable by computable monotonous betting. In order to see this, let z_n, o_n be the number of 0s and 1s respectively, in the first n bits of Ville's sequence, so that $z_n \geq o_n$ for all n. In the case where $\sup_n(z_n - o_n) = \infty$ our strategy is to start with capital 1, and bet wager 1 on outcome 0 at each step. In the case where $\limsup_n(z_n - o_n) := k < \infty$, given k and a stage t such that for all $n \geq t$ we have $z_n - o_n \leq k$, we can use the following strategy: given any stage $s_0 > t$, find some $n \geq s_0$ such that $z_n - o_n = k$ and at this n bet on 1. In order to avoid the dependence of this martingale on the parameters k, t, we can consider a mixture including a martingale for each possible pair (k, t), with initial capital for the s-th martingale equal to 2^{-s}. (so that the total initial capital is finite.) In this case, the mixture is still a computable martingale. Note that in the first case the martingale is 0-sided and in the second case it is 1-sided; moreover in both cases, under the respective assumption, the martingales are successful on Ville's sequence. The mixture of these two strategies is a computable separable martingale and is successful on Ville's sequence.

Success on skewed sequences

Now we show that given a sequence X with limiting frequency of 0s different than 1/2, there is a computable single-sided betting strategy that is successful on X. Moreover there is a separable martingale which succeeds on every such X, irrespective of whether the frequency is above or below 1/2, or even how much it differs from 1/2. A slightly more general version of these facts, is a result of the following form of Hoeffding's Inequality, which we prove below via betting strategies, as it will be used in our later arguments as well.

Lemma 5.2.1 (Hoeffding's Inequality for prediction functions)**.** *Given $\epsilon > 0$, $n \in \omega$ and a prediction function f, there are at most $r_\epsilon^{-n} \cdot 2^n$ many strings of length n along which the f-guess correct rates are no less than $1/2 + \epsilon$, where $r_\epsilon > 1$ is a function of ϵ. So there are at least $(1 - 2r_\epsilon^{-n}) \cdot 2^n$ many strings of length n along which the f-guess correct rates are in the interval $(1/2 - \epsilon, 1/2 + \epsilon)$.*

Proof. Given f, we just need to prove the first statement, because it implies that there are at most $r_\epsilon^{-n} \cdot 2^n$ many strings of length n along which the $(1-f)$-guess correct rates are no less than $1/2+\epsilon$, which also means the f-guess correct rates are no greater than $1/2-\epsilon$. Thus, the second statement in the lemma follows.

Our proof idea is to find a betting strategy which is successful along any sequence whose f-guess correct rate deviates from $1/2$. For each σ let p_σ be the f-guess correct rate along σ. For each $\epsilon > 0$ let $q = 1/2+\epsilon$, we define a function $d : 2^{<\omega} \mapsto \mathbb{R}$ by letting $d(\sigma) = 2^{|\sigma|} \cdot q^{p_\sigma|\sigma|} \cdot (1-q)^{(1-p_\sigma)|\sigma|}$. Then for all $\sigma \in 2^{<\omega}$ we have $d(\sigma) > 0$ and

$$\begin{aligned} d(\sigma^\frown 0) + d(\sigma^\frown 1) &= 2^{|\sigma|+1} \cdot \left(q^{p_\sigma|\sigma|+1} \cdot (1-q)^{(1-p_\sigma)|\sigma|} + q^{p_\sigma|\sigma|} \cdot (1-q)^{(1-p_\sigma)|\sigma|+1}\right) \\ &= 2^{|\sigma|+1} \cdot q^{p_\sigma|\sigma|} \cdot (1-q)^{(1-p_\sigma)|\sigma|} = 2d(\sigma). \end{aligned}$$

So d is a martingale which bets $d(\sigma^\frown f(\sigma)) - d(\sigma) = (2q-1)d(\sigma)$ on $f(\sigma)$ at stage σ.

By considering the derivative, we can see that the function $x \mapsto x^x \cdot (1-x)^{1-x}$ in $x \in (0,1)$ takes its minimum value $1/2$ at $x = 1/2$. Let $r_\epsilon = 2(1/2+\epsilon)^{1/2+\epsilon} \cdot (1/2-\epsilon)^{1/2-\epsilon}$, then as $\epsilon > 0$ we get $r_\epsilon > 1$.

Let $T_n = \{\sigma \in 2^n : p_\sigma \geq q\}$, then for each $\sigma \in T_n$

$$d(\sigma) = \left(2 \cdot q^{p_\sigma} \cdot (1-q)^{1-p_\sigma}\right)^n \geq \left(2 \cdot q^q \cdot (1-q)^{1-q}\right)^n = r_\epsilon^n,$$

where the inequality holds because by considering the derivative the function $x \mapsto q^x(1-q)^{1-x}$ is increasing in $x \in (0,1)$ when $q > 1/2$. Then from Kolmogorov's Inequality of Theorem 4.3.3 it follows that

$$1 = d(\lambda) \geq \sum_{\sigma \in T_n} 2^{-|\sigma|} \cdot d(\sigma) \geq |T_n| \cdot 2^{-n} \cdot r_\epsilon^n.$$

Thus, $|T_n| \leq r_\epsilon^{-n} \cdot 2^n$ as required. □

As $r_\epsilon > 1$, r_ϵ^{-n} goes to 0 when n goes to infinity. Then Lemma 5.2.1 says that for each total prediction function f, for long enough strings with high probability the f-guess correct rate along it is near $1/2$.

In fact, there exists a separable computable martingale which succeeds on every sequence X with the property that the proportion of correct f-guesses along X does not reach limit 1/2. For each $q \in (1/2, 1)$ define $T_q(\sigma) = 2^{|\sigma|} \cdot q^{z_\sigma} \cdot (1-q)^{o_\sigma}$ where z_σ is the number of correct f-guesses along σ and o_σ is the number of false f-guesses along σ. By the proof of Lemma 5.2.1, $T_q(\sigma)$ is a martingale and $\limsup_n T_q(X \restriction n) = \infty$ for each X such that $\limsup_n z_{X\restriction n}/n > q$. Similarly, $T_q(\sigma)$ is a martingale for each $q < 1/2$, and $\limsup_n T_q(X \restriction n) = \infty$ for each X such that $\limsup_n z_{X\restriction n}/n < q$. Let $q_i = 1/2 + 2^{-i-1}$ and $p_i = 1/2 - 2^{-i-1}$ for each i and define:

$$N(\sigma) = \sum_i 2^{-i} \cdot T_{q_i}(\sigma) + \sum_i 2^{-i} \cdot T_{p_i}(\sigma).$$

Then N is a computable martingale and by the properties of T_{q_i}, T_{p_i}, it succeeds on every X for which the proportion of correct f-guesses does not tend to 1/2. In the case that f is the constant zero function all T_{q_i} are 0-sided and all T_{p_i} are 1-sided. Then N is a computable separable martingale, which succeeds on every X for which the frequency of 0 in the initial segments does not tend to 1/2.

Decomposition of a martingale

Lemma 5.2.2. *Every martingale M is the product of a 0-sided martingale N and a 1-sided martingale T. Moreover N, T are computable from M.*

Recall that we use c_M to denote the betting coefficient of a (super)martingale M.

Proof. Suppose M is a martingale. We define two betting coefficients c_N and c_T as follows. Let $c_N(\lambda) = c_T(\lambda) = 1$. For all $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$, let

$$\begin{array}{llll} c_N(\sigma^\frown i) = c_M(\sigma^\frown i) & \text{and} & c_T(\sigma^\frown i) = 1 & \text{if } c_M(\sigma^\frown 0) > 1; \\ c_N(\sigma^\frown i) = 1 & \text{and} & c_T(\sigma^\frown i) = c_M(\sigma^\frown i) & \text{if } c_M(\sigma^\frown 0) \leq 1. \end{array}$$

For all $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$, as by Proposition 4.1.6 we have $c_M(\sigma^\frown 0) + c_M(\sigma^\frown 1) = 2$, it holds that

$$c_N(\sigma^\frown 0) + c_N(\sigma^\frown 1) = c_T(\sigma^\frown 0) + c_T(\sigma^\frown 1) = 2. \tag{5.5}$$

And obviously,

$$c_M(\sigma) = c_N(\sigma) \cdot c_T(\sigma). \tag{5.6}$$

Let $N(\sigma) = M(\lambda) \cdot \prod_{\tau \preceq \sigma} c_N(\tau)$ and $T(\sigma) = \prod_{\tau \preceq \sigma} c_T(\tau)$. Then by (5.5) N and T are martingales and clearly, they are computable from M.

On the other hand, by (5.6), $M(\sigma) = N(\sigma) \cdot T(\sigma)$ for all $\sigma \in 2^{<\omega}$. □

The following thoerem and corollary are direct consequences of Lemma 5.2.2.

Theorem 5.2.3 (Barmpalias, Fang, and Lewis-Pye [14]). *Given a computable martingale M, there exist a computable separable martingale N superior to M.*

Corollary 5.2.4. $\mathsf{Succ}[\mathit{sCM}] = \mathsf{Succ}[\mathit{CM}]$.

5.3 Strongly Left-C.E. Separable Supermartingales

Theorem 5.2.3 says that, in terms of computable martingales, the collection of all successful sequences of an arbitrary strategy can be covered by a separable one. Now we explore the situation for strongly left-c.e. supermartingales.

Success on sequences with dimension less than 1/2

Theorem 5.3.1 (Barmpalias, Fang, and Lewis-Pye [14]). *There exists a strongly left-c.e. separable martingale M which succeeds on all sequences with dimension less than 1/2.*

Proof. Let $\{V_i\}_{i\in\omega}$ be a universal 1/2-test. For each $\sigma \in 2^{<\omega}$, we define a 0-sided martingale N_σ and a 1-sided martingale T_σ. N_σ starts with initial capital $2^{-|\sigma|/2}$ and bets all capital on all the 0s along σ, while placing no bets on other strings. Formally, $N_\sigma(\lambda) = 2^{-|\sigma|/2}$ and for every $\rho \succ \lambda$ let

$$N_\sigma(\rho) = \begin{cases} 2 \cdot N_\sigma(\rho^-) & \text{if } \rho^- {}^\frown 0 \preceq \sigma \wedge \rho = \rho^- {}^\frown 0, \\ 0 & \text{if } \rho^- {}^\frown 0 \preceq \sigma \wedge \rho = \rho^- {}^\frown 1, \\ N_\sigma(\rho^-) & \text{otherwise.} \end{cases}$$

By definition for each $\rho \succeq \sigma$ we have $N_\sigma(\rho) = 2^{z_\sigma - |\sigma|/2}$, where z_σ denotes the number of 0s in σ. Hence $N_\sigma(\rho) \geq 1$ for all $\rho \succeq \sigma$ whenever at least half of the bits of σ are 0s. The definition and properties of T_σ is analogous, except that it bets all capital on all the 1s along σ. Let

$$N = \sum_i \sum_{\sigma \in V_i} N_\sigma \quad , \quad T = \sum_i \sum_{\sigma \in V_i} T_\sigma \quad \text{and} \quad M = N + T.$$

By the measure properties of $\{V_i\}_{i\in\omega}$ the initial capital of N, T is finite. Since each N_σ is computable 0-sided, N is strongly left-c.e. 0-sided, and in the same way T is strongly left-c.e. 1-sided. Thus, M is strongly left-c.e. separable. Moreover if there are at least k many members of $\{V_i\}_{i\in\omega}$ containing a prefix of X, there exists $j \in \{0, 1\}$ such that at least half of these prefixes have at least half of their digits equal to j. Hence there exists n such that, according to whether j is 0 or 1, we have $N(\tau) \geq k/2$ or $T(\tau) \geq k/2$ respectively for each $\tau \succeq X \upharpoonright n$. Given X with $\dim X < 1/2$, by (4.15) it is not weakly 1/2-random. Then by the universality of $\{V_i\}_{i\in\omega}$, X has prefixes in infinitely many V_i. It follows that $\lim_n N(X \upharpoonright n) = \infty$ or $\lim_n T(X \upharpoonright n) = \infty$. In any case we have $\lim_n M(X \upharpoonright n) = \infty$ as required. □

Theorem 5.3.1 shows that no sequence of effective dimension less than 1/2 is *sSLM*-random. However, we will see below that there exits an *sSLM*-random sequence with effective dimension 1/2. The proof idea will be used in next section to prove the more generalized results about *dSLM*.

Failure on a sequence with dimension 1/2

Theorem 5.3.2 (Barmpalias, Fang, and Lewis-Pye [14]). *Given a computable prediction function f, for any strongly left-c.e. f-sided supermartingale M, there exists a sequence X with dimension 1/2 on which M does not succeed.*

Given a strongly left-c.e. f-sided supermartingale M, let $\{M_i\}$ be a canonical approximation to M. Without loss of generality we assume that $M(\lambda) \leq 2^{-1}$. Let us define $\hat{M}(\rho) = \max\{M(\tau) \colon \tau \preceq \rho\}$. Our proof idea is to construct a series of strings $\{\sigma_n\}_{n\in\omega}$ such that $\sigma_0 = \lambda$, $\sigma_n \prec \sigma_{n+1}$ and

$$\forall n \ \hat{M}(\sigma_n) \leq 1 - 2^{-n-1}. \tag{5.7}$$

Then if we define $X = \lim_n \sigma_n$, we get $\limsup_n M(X \upharpoonright n) \leq 1$, i.e. $X \notin \mathsf{Succ}(M)$. On the other hand, during the construction we will also construct a KC set (bounded request set) V by enumerating $\langle q_n \cdot |\sigma_n|,\ \sigma_n\rangle$ into V for every newly (re)defined σ_n, where $q_n = 1/2 + 2^{-n-1}$. Thus, by Corollary 2.1.8 of

KC Theorem,

$$\mathrm{K}(\sigma_n) \le q_n \cdot |\sigma_n| + \mathbf{O}(1)\,. \tag{5.8}$$

Then clearly the effective Hausdorff dimension of X is no greater than $1/2$. With Theorem 5.3.1 it imply that $\dim(X) = 1/2$.

At stage s supposing inductively that σ_n has been determined, to issue a candidate for σ_{n+1} we will try to find some string extending σ_n such that

$$M_s(\tau) \le M_s(\sigma_n) + 2^{-n-3} \text{ for all } \sigma_n \prec \tau \preceq \sigma_{n+1}. \tag{5.9}$$

As M is a left-c.e. supermartingale, in order to keep (5.7) true, we might need to change the approximation to σ_{n+1} a number of times. Suppose stage t is the next stage where $\hat{M}_t(\sigma_{n+1}) > 1 - 2^{-n-2}$ and we need change the approximation to σ_{n+1}. However, as σ_n is still safe, so $\hat{M}_t(\sigma_n) \le 1 - 2^{-n-1}$ holds. Then there is $\sigma_n \prec \tau \preceq \sigma_{n+1}$ such that $M_t(\tau) > 1 - 2^{-n-2}$. By (5.9) we have $M_s(\tau) \le M_s(\sigma_n) + 2^{-n-3} \le 1 - 2^{-n-1} + 2^{-n-3}$, then $M_t(\tau) - M_s(\tau) > 2^{-n-3}$. That is to say, our construction tolerates an increase of $M(\tau)$ by $\delta_n = 2^{-n-3}$ for all $\sigma_n \prec \tau \preceq \sigma_{n+1}$ at later stages. This tolerance will be essential for us to limit the changing times of the approximation. For convenience we make the following definition.

Definition 5.3.3. Given a prediction function f, $\sigma \in 2^{<\omega}$, $\epsilon \in \{0,1\}$, $\ell \in \mathbb{N}$, let $S(f,\sigma,\epsilon,\ell)$ be the set of strings extending σ of length $|\sigma| + \ell$ along which the number of correct f-guesses after σ is less than $(1/2+\epsilon)\ell$.

Note that the set $S(f,\sigma,\epsilon,\ell)$ is computable from f,σ,ϵ,ℓ. The property of the strings in the set $S(f,\sigma,\epsilon,\ell)$ ensures the following fact.

Lemma 5.3.4. *Let M be a left-c.e. f-sided supermartingale with canonical approximation $\{M_i\}$. Given $\sigma \in 2^{<\omega}$, $\epsilon \in \{0,1\}$, $\ell \in \mathbb{N}$, and $\rho \in S(f,\sigma,\epsilon,\ell)$, for all $t > s$ if there exist $\sigma \prec \tau \preceq \rho$ such that $M_t(\tau) - M_s(\tau) > \delta$, then it must be that $M_t(\sigma) - M_s(\sigma) > \delta \cdot 2^{-(1/2+\epsilon)\ell}$.*

Proof. Fix $t > s$. Let $N = M_t - M_s$. As $\{M_i\}$ is a canonical approximation to M, N is also an f-sided supermartingale. As the number of correct f-guesses after σ along ρ is less than $(1/2+\epsilon)\ell$, then $N(\tau) \le N(\sigma) \cdot 2^{(1/2+\epsilon)\ell}$ for all $\sigma \prec \tau \preceq \rho$. Thus, if there exist $\sigma \prec \tau \preceq \rho$ such that $N(\tau) = M_t(\tau) - M_s(\tau) > \delta$ then $M_t(\sigma) - M_s(\sigma) = N(\sigma) \ge N(\tau) \cdot 2^{-(1/2+\epsilon)\ell} > \delta \cdot 2^{-(1/2+\epsilon)\ell}$. □

Let us set two series of parameters ϵ_n and ℓ_n which will be fixed later. Our candidates of σ_{n+1} will be chosen from $S_n = S(f, \sigma_n, \epsilon_n, \ell_n)$. Then from any stage on if there exist $\sigma_n \prec \tau \preceq \sigma_{n+1}$ such that $M(\tau)$ increases by δ_n, at the same time $M(\sigma_n)$ should have already increased by at least $\delta_n \cdot 2^{-(1/2+\epsilon_n)\ell_n}$. Assuming σ_n is stable at some stage, by (5.7), $M(\sigma_n)$ is bounded above by 1. Hence the approximation to σ_{n+1} will change at most $p_n = 2^{(1/2+\epsilon_n)\ell_n}/\delta_n$ many times from that stage on. That is to say, the approximation to σ_{n+1} changes at most $\prod_{0\le i\le n} p_i$ many times. As required by (5.8), we need to keep the total measure of the requests in V bounded above by 1. It suffices to keep the total measure of the requests enumerated for the approximations of σ_{n+1} bounded above by 2^{-n-1}. For this, we set the following requirement for our parameters:

$$2^{-q_{n+1}\cdot\sum_{0\le i\le n}\ell_i} \cdot \prod_{0\le i\le n} p_i \le 2^{-n-1}. \tag{5.10}$$

On the other hand, the following lemma ensures that if the set S_n is large enough, a string satisfying (5.9) can always be found in S_n.

Lemma 5.3.5. *Given any supermartingale M, $\sigma \in 2^{<\omega}$, $\epsilon \in (0,1)$ and $S \subseteq \{\tau \in 2^{|\sigma|+\ell} \colon \tau \succ \sigma\}$ with $|S| \ge (1-\epsilon)\cdot 2^{\ell}$, there exists some $\tau \in S$ such that $M(\rho) \le M(\sigma)/(1-\epsilon)$ for all $\sigma \prec \rho \preceq \tau$.*

Proof. Towards a contradiction suppose that there exists no such string in S. For each $\tau \in S$ let τ^* be the shortest initial segment of τ extending σ such that $M(\tau^*) > M(\sigma)/(1-\epsilon)$. Then $S^* = \{\tau^* \colon \tau \in S\}$ is a prefix-free set of strings. Since every element of S has an initial segment in S^* it follows that:

$$\begin{aligned}
\sum_{\tau^*\in S^*} 2^{-(|\tau^*|)} \cdot M(\tau^*) &> \frac{M(\sigma)}{1-\epsilon} \cdot \sum_{\tau^*\in S^*} 2^{-|\tau^*|} \\
&\ge \frac{M(\sigma)}{1-\epsilon} \cdot \sum_{\tau\in S} 2^{-|\tau|} \\
&\ge \frac{M(\sigma)}{1-\epsilon} \cdot (1-\epsilon)\cdot 2^{\ell} \cdot 2^{-(|\sigma|+\ell)} \\
&= 2^{-|\sigma|} \cdot M(\sigma)
\end{aligned}$$

which contradicts the Kolmogorov's Inequality in Theorem 4.3.3. □

Moreover, by the following lemma, we can choose ℓ_n large enough so that $|S_n| \ge (1-\epsilon_n)\cdot 2^{\ell_n}$.

σ_n	the nth initial segment of X with approximations $\sigma_n[s]$
ℓ_n	length difference of σ_{n+1} and σ_n, set as (5.12)
ϵ_n	value such that the number of correct f-guesses after σ_n along σ_{n+1} is no more than $(1/2+\epsilon_n)\ell_n$ and $\lvert S_n\rvert \geq (1-\epsilon_n)2^\ell$, set as 2^{-n-3}
S_n	candidates set for σ_{n+1}, set as $S(f, \sigma_n, \epsilon_n, \ell_n)$
q_n	bound for $\mathrm{K}(\sigma_n)/\lvert\sigma\rvert$, set as $1/2 + 2^{-n-1}$
δ_n	tolerance of increase of $M(\tau)$ for all $\sigma_n \prec \tau \preceq \sigma_{n+1}$, equals 2^{-n-3}
p_n	maximal σ_{n+1} change times after σ_n is settled, equals $2^{(1/2+\epsilon_n)\ell_n+n+3}$
h_n	lower bound for ℓ_n to get $\lvert S_n\rvert$ large enough, set as $\lceil -\log\epsilon_n/\log r_{\epsilon_n}\rceil$

Table 5.2 Parameters for the construction of Theorem 5.3.2

Lemma 5.3.6. *$|S(f,\sigma,\epsilon,\ell)| \geq (1-r_\epsilon^{-\ell})\cdot 2^\ell$, where $r_\epsilon > 1$ is a function of ϵ.*

Proof. Let f' be the prediction function defined as $f'(\tau) = f(\sigma \frown \tau)$ for all $\tau \in 2^{<\omega}$. Then $S(f', \lambda, \epsilon, \ell) = S(f, \sigma, \epsilon, \ell)$. By Lemma 5.2.1, we have $|S(f,\sigma,\epsilon,\ell)| = |S(f',\lambda,\epsilon,\ell)| \geq (1-r_\epsilon^{-\ell})\cdot 2^\ell$, where $r_\epsilon > 1$ is a function of ϵ. □

Let $h_n = \lceil -\log\epsilon_n/\log r_{\epsilon_n}\rceil$, which implies $r_{\epsilon_n}^{-\ell} \leq \epsilon_n$ for all $\ell \geq h_n$. By Lemmas 5.3.5 and 5.3.6, if we set ϵ_n and ℓ_n such that

$$\epsilon_n = 2^{-n-3} \text{ and } \ell_n \geq h_n \tag{5.11}$$

then $|S_n| \geq (1-\epsilon_n)\cdot 2^{\ell_n}$. In case σ_n satisfies (5.7) at stage s, there exists $\tau \in S_n$ such that for all $\sigma_n \prec \rho \preceq \tau$ we have $M_s(\rho) \leq M_s(\sigma_n)/(1-\epsilon_n) \leq M_s(\sigma_n)+2^{-n-3}$ which satisfies (5.9).

Combine (5.10) and (5.11), and put in the value of other parameters, we get

$$\ell_n \geq 2^{n+2}\cdot(n+1)(n+8) + \sum_{0\leq i<n}(2^{n-i}-2)\ell_i.$$

Thus, it suffices to define ℓ_n inductively as follows,

$$\ell_n = \max\left\{2^{n+2}\cdot(n+1)(n+8) + \sum_{0\leq i<n}(2^{n-i}-2)\ell_i\ ,\ h_n\right\}. \tag{5.12}$$

Proof of Theorem 5.3.2. For all n we set the parameters $\epsilon_n, \ell_n, q_n, S_n$ as in previous discussion, which are summarized in Table 5.2. We inductively define the approximations $\sigma_n[s]$ of σ_n for all n, in stages s. At stage $s+1$ we say the segment σ_n *requires attention* if either $\sigma_n[s]\uparrow$, or $\sigma_n[s]\downarrow$ and $\hat{M}_{s+1}(\sigma_n[s]) > 1-2^{-n-1}$.

Construction:

Let $\sigma_0[s]=\lambda$ for all s and $\sigma_n[0]\uparrow$ for all $n>0$. During the construction, at any stage all unmentioned σ_n remain unchanged.

At stage $s+1$: Find the least number $n \le s$ such that σ_{n+1} requires attention. Then check and do the following.

- If $\sigma_{n+1}[s]\uparrow$, define $\sigma_{n+1}[s+1]$ to be the leftmost string ρ in S_n such that:

  $$M_{s+1}(\tau) \le M_{s+1}(\sigma_n[s]) + 2^{-n-3} \text{ for all } \sigma_n[s] \prec \tau \preceq \rho. \tag{5.13}$$

  and enumerate a request $\langle q_{n+1}\cdot|\sigma_{n+1}|,\ \sigma_{n+1}[s+1]\rangle$ into V.
- if $\sigma_{n+1}[s]\downarrow$, set $\sigma_i[s+1]\uparrow$ for all $i \ge n+1$.

Otherwise go to the next stage directly.

Verification:

By Lemma 5.3.5 and 5.3.6, and by the discussion to follow, during the construction a string ρ satisfying (5.13) can always be found. Thus, the construction is well defined.

Lemma 5.3.7. *Every σ_n only requires attention for finitely many times.*

Proof. Let $k \ge 0$. Inductively, suppose for all $i<k$, σ_i only requires attention for finitely many times. We show that σ_k requires attention for finitely many times. By assumption there is a stage s_0 from which on all $\sigma_i, i<k$ do not require attention. From stage s_0 on, whenever σ_k requires attention it will be treated at that stage, either from defined to undefined or from undefined to defined. Every two such stages result in a change of the approximation to σ_k. By our discussion above, from stage s_0 on, the approximation to σ_k only changes at most p_{k-1} many times. Thus, it requires attention for only finitely many times, which completes our proof. □

By Lemma 5.3.7, for each n there is a stage after which σ_n never requires attention. That is to say, σ_n converges and $\hat{M}(\sigma_n) \le 1-2^{-n-1}$ holds. On the other hand, by the configuration of our parameters, the total measure of the requests enumerated into V is bounded by 1. Thus, both (5.7) and (5.8) hold, which completes our proof. $\square$

A simple adaption the above proof leads us to the following theorem. The only attention we need to pay is that now the set S_n is defined as $S(0, \sigma_n, \epsilon_n, \ell_n) \cap S(1, \sigma_n, \epsilon_n, \ell_n)$. Then the value of h_n needs to be redefined to make sure $|S_n| \geq (1 - \epsilon_n) \cdot 2^{\ell_n}$ still holds for $\ell_n \geq h_n$. This can be easy achieved as $|S_n| \geq (1 - 2r_{\epsilon_n}^{-\ell_n}) \cdot 2^{\ell_n}$.

Theorem 5.3.8 (Barmpalias, Fang, and Lewis-Pye [14])**.** *For any strongly left-c.e. separable supermartingale M, there exists a sequence X such that* $\dim(X) = 1/2$ *and* $X \notin \mathsf{Succ}(M)$.

Remember that all Martin-Löf random sequences have dimension 1, and there is an optimal strongly left-c.e. supermartingale which will succeed on all non-Martin-Löf random sequences, which include all sequences with dimension less than 1. Also remember that there is a universal strongly left-c.e. separable supermartingale, so from Theorem 5.3.8 we get the following corollaries directly. It is instructive to contrast Corollary 5.3.9 with Theorem 5.2.3.

Corollary 5.3.9. *There exist a strongly left-c.e. supermartingale M, such that* $\mathsf{Succ}(N) \subsetneq \mathsf{Succ}(M)$ *for all strongly left-c.e. separable supermartingale N.*

Corollary 5.3.10. $\mathsf{Succ}[\mathit{sSLS}] \neq \mathsf{Succ}[\mathit{SLS}] = \mathsf{Succ}[\mathit{LM}]$.

5.4 Strongly Left-C.E. Decidably-sided Supermartingales

Given the fact that there is no optimal decidably-sided supermartingale, from Theorem 5.3.2 we still cannot conclude that $\mathsf{Succ}[\mathit{dSLS}] \neq \mathsf{Succ}[\mathit{SLS}]$. However, this can be proved by an adaption and generalization of the proof for Theorem 5.3.2.

Theorem 5.4.1 (Barmpalias, Fang, and Lewis-Pye [14])**.** *There exists a sequence X with dimension 1/2 such that no decidably-sided supermartingale succeeds on it.*

In the proof of Theorem 5.3.2, as the KC set V needs to be c.e. , our construction is computable, especially, all the sets S_n are computable, because that is where the candidates for σ_{n+1} are chosen. Although Lemma 5.3.6 also holds for partial predication functions, the set $S(f, \sigma, \epsilon, \ell)$ is no longer computable for a partial computable function. Thus, here we opt for a less constructive initial segment argument, which uses the facts we obtained in § 5.3 in a modular way.

First we prove the following lemma, which will be the main tool for the proof of Theorem 5.4.1. Let $q_n = 1/2 + 2^{-n-3}$.

Lemma 5.4.2. *There exists a prefix-free machine Q such that for every $n \in \mathbb{N}, \sigma \in 2^{<\omega}$, and $M = \sum_{i \leq n} N_i$ where each $N_i, i \leq n$ is a strongly left-c.e. f_i-sided supermartingale for some total computable prediction function f_i, if*

$$\hat{M}(\sigma) \leq 1 - 2^{-n-1}. \tag{5.14}$$

then there exists $\tau \succ \sigma$ such that $\mathrm{K}_Q(\tau) \leq q_n \cdot |\tau|$ and

$$\hat{M}(\tau) \leq 1 - 3 \cdot 2^{-n-3} \tag{5.15}$$

Proof. To construction such a prefix-free machine Q, we enumerate a KC set V. Note that V needs to be a c.e. set. As there is an effective list of canonical computable approximations to all left-c.e. supermartingales with initial capital less than 1, let H be an effective list of all possible tuples of $n \in \mathbb{N}, \sigma \in 2^{<\omega}$, $\{f_i\}_{i \leq n}$ and $\{N_i[s]\}_{i \leq n}$, where each $f_i, i \leq n$ is a partial computable prediction function, each $\{N_i[s]\}_{s \in \omega}, i \leq n$ is a canonical computable approximations to some left-c.e. supermartingales with initial capital less than 1. We will define an effective map which takes as an input $\eta = \langle n, \sigma, \{f_i\}_{i \leq n}, \{N_i[s]\}_{i \leq n} \rangle \in H$ and always outputs a *sufficiently small* part V_η of V (dealing with the specific input η) and an approximation $\tau[s]$ such that if η meets the hypothesis of Lemma 5.4.2 then $\tau[s]$ converges to some τ which satisfies the requirements of the lemma. Let $g \colon H \mapsto \mathbb{N}$ a one-to-one computable function so that $\sum_{\eta \in H} 2^{-g(\eta)} < 1$. Then to make sure that V is a KC set, i.e. $\mu(V) \leq 1$ it suffices to make sure $\mu(V_\eta) \leq 2^{-g(\eta)}$ for each $\eta \in H$.

Now we construct the effective map $\eta \mapsto (V_\eta, \tau[s])$. It is an adaption of the construction for Theorem 5.3.2 As now the gap between $\hat{M}(\sigma)$ and $\hat{M}(\tau)$ is

2^{-n-3}, we will find a candidate τ such that

$$\hat{M}(\tau) \leq 1 - 7 \cdot 2^{-n-4} \tag{5.16}$$

Thus, the tolerance to an increase of $\hat{M}(\tau)$ is now $\delta_n = 2^{-n-4}$. In the same way as before, we will find the candidates for τ in a collection S_n of strings extending σ of length $|\sigma| + \ell_n$ such that the growth potential is limited. Assume $f_i, i \leq n$ are total functions and each $N_i, i \leq n$ is a strongly left-c.e. f_i-sided supermartingale. We define $S_n = \bigcap_{i \leq n} S(f_i, \sigma, \epsilon_n, \ell_n)$, so that Lemma 5.3.4 also holds for the supermartingale $M = \sum_{i \leq n} N_i$. Hence the approximation to τ will change at most $p_n = 2^{(1/2+\epsilon_n)\ell_n}/\delta_n$ many times. Then to keep $\mu(V_\eta) \leq 2^{-g(\eta)}$, we have the following requirement for our parameters:

$$2^{-q_n \cdot (\ell_n + |\sigma|)} \cdot p_n \leq 2^{-g(\eta)}. \tag{5.17}$$

On the other hand, by Lemma 5.3.6, $|S(f_i, \sigma, \epsilon_n, \ell_n)| \geq (1 - r_{\epsilon_n}^{-\ell_n}) \cdot 2^{\ell_n}$ holds for each $i \leq n$. Thus, $|S_n| = |\bigcap_{i \leq n} S(f_i, \sigma, \epsilon_n, \ell_n)| \geq (1 - (n+1) \cdot r_{\epsilon_n}^{-\ell_n}) \cdot 2^{\ell_n}$. Now define $h_n = \lceil (\log(n+1) - \log \epsilon_n) / \log r_{\epsilon_n} \rceil$. Then by setting

$$\epsilon_n = 2^{-n-4} \text{ and } \ell_n \geq h_n \tag{5.18}$$

we also get $|S_n| \geq (1 - \epsilon_n) \cdot 2^{\ell_n}$. By Lemma 5.3.5 and (5.14) there exists $\tau \in S_n$ such that for all $\sigma \prec \rho \preceq \tau$ we have $M(\rho) \leq M(\sigma)/(1 - \epsilon_n) \leq M(\sigma) + 2^{-n-4} \leq 1 - 2^{-n-1} + 2^{-n-4} = 1 - 7 \cdot 2^{-n-4}$. As $\hat{M}(\sigma) \leq 1 - 2^{-n-1}$, then $\hat{M}(\tau) \leq 1 - 7 \cdot 2^{-n-4}$. This means a candidate for τ satisfying (5.16) can always be found in S_n. Combine (5.17) and (5.18), and put in the value of other parameters, we get the following appropriate value for ℓ_n:

$$\ell_n = \max\left\{2^{n+4} \cdot (g(\eta) + n + 4) - 2|\sigma| \ , \ h_n\right\}. \tag{5.19}$$

$\boxed{\textbf{Construction of the map } \eta \mapsto (V_\eta, \tau[s])\textbf{:}}$

Given $\eta = \langle n, \sigma, \{f_i\}_{i \leq n}, \{N_i[s]\}_{i \leq n} \rangle \in H$, let $M_s = \sum_{i \leq n} N_i[s]$. We set the parameters $\epsilon_n, \ell_n, q_n, S_n$ as in previous discussion, which are summarized in Table 5.3. We define the approximations $\tau[s]$ of τ, in stages s. At stage $s+1$ we say τ *requires attention* if either $\tau[s] \uparrow$, or $\tau[s] \downarrow$ and $\hat{M}_{s+1}(\tau[s]) > 1 - 3 \cdot 2^{-n-3}$.

Let $\tau[0] \uparrow$ and $V_\eta[0] = \emptyset$. Suppose $N_i[-1] = 0$ for all $i \leq n$.

ℓ_n	length difference of τ and σ, set as (5.19)
ϵ_n	value such that the number of each correct f_i-guesses after σ along τ is no more than $(1/2+\epsilon_n)\ell_n$ and $\lvert S_n\rvert \geq (1-\epsilon_n)2^{\ell}$, set as 2^{-n-4}
S_n	candidates set for τ, set as $\bigcap_{i\leq n} S(f_i, \sigma, \epsilon_n, \ell_n)$
q_n	bound for $\mathrm{K}(\tau)/\lvert\tau\rvert$, set as $1/2 + 2^{-n-3}$
δ_n	tolerance of increase of $M(\tau)$ for all $\sigma \prec \tau \preceq \tau$, equals 2^{-n-4}
p_n	maximal change times for τ, equals $2^{(1/2+\epsilon_n)\ell_n+n+4}$
h_n	one lower bound for ℓ_n, set as $\lceil (\log(n+1) - \log \epsilon_n)/\log r_{\epsilon_n} \rceil$

Table 5.3 Parameters for the construction of the map $\eta \mapsto (V_\eta, \tau[s])$

At stage $s+1$: If for each $i \leq n$, $f_i[s+1]$ is defined on all strings in $2^{\leq(|\sigma|+\ell_n)}$ and each $N_i[j+1] - N_i[j], -1 \leq j \leq s$ is f_i-sided, and τ requires attention, check and do the following.

- If $\tau[s]\uparrow$, define $\tau[s+1]$ to be the leftmost string ρ in S_n such that:

$$\hat{M}_{s+1}(\rho) \leq 1 - 7 \cdot 2^{-n-4}. \tag{5.20}$$

  and enumerate a request $\langle q_n \cdot |\tau|,\ \tau[s+1]\rangle$ into V_η.

- If $\tau[s]\downarrow$, set $\tau[s+1]\uparrow$.

Otherwise go to the next stage directly.

Verification of the map $\eta \mapsto (V_\eta, \tau[s])$:

In case there is no such a stage s such that for each $i \leq n$, $f_i[s]$ is defined on all strings in $2^{\leq(|\sigma|+\ell_n)}$ and each $N_i[j+1] - N_i[j], -1 \leq j < s$ is f_i-sided, τ will be never defined and we will never issue any code into V_η, thus, $\mu(V_\eta) = 0 \leq 2^{-g(\eta)}$. In case there is such stage s, then by our configuration of the parameters and the discussion above, during the construction the string ρ satisfying (5.20) can always be found. Thus, the construction is well defined. Also following from our configuration of the parameters and the discussion above, the approximation to τ only changes at most p_n many times and $\mu(V_\eta) \leq 2^{-g(\eta)}$ is also ensured. As from stage s on, whenever τ requires attention it will be treated at that stage, either from defined to undefined or from undefined to defined. Every two such stages result in a change of the approximation to τ. Thus, there is a stage after which σ never requires attention. Then by construction τ converges and $\hat{M}(\tau) \leq 1 - 3 \cdot 2^{-n-3}$ holds.

For the proof the Lemma 5.4.2, note that whenever we are given m, σ and M as stated in the lemma, there is some $\eta = \langle n, \sigma, \{f_i\}_{i \leq n}, \{N_i[s]\}_{i \leq n} \rangle \in H$ such that for each $i \leq n$, f_i is a total computable prediction function, $N_i[s]$ is a canonical approximation to a strongly left-c.e. f_i-sided supermartingale N_i and $M = \sum_{i \leq n} N_i$. Then our map $\eta \mapsto (V_\eta, \tau[s])$ gives the τ as required by the lemma. This completes the proof of Lemma 5.4.2. □

Proof of Theorem 5.4.1. Let $\{M_i\}_{i \in \omega}$ be a (non-effective) list of all strongly left-c.e. decidably-sided supermartingales with initial capital less than 1 and with canonical computable approximations $\{M_i[s]\}_{i \in \omega}$. Thus, to prove Theorem 5.4.1 it suffices to construct a sequence X such that $\dim(X) = 1/2$ and $X \notin \mathsf{Succ}(M_i)$ for all i.

Construction:

Let $\sigma_0 = \lambda$ and Q be the machine from Lemma 5.4.2. For each $n \geq 0$, inductively define $D_n \colon = \sum_{i \leq n} 2^{-|\sigma_i| - i - 2} \cdot M_i$, and assume that

$$\hat{D}_n(\sigma_n) \leq 1 - 2^{-n-1}. \tag{5.21}$$

Then let σ_{n+1} be an extension of σ_n given by Lemma 5.4.2 with $\sigma = \sigma_n$ and $M = D_n$. Define $X = \lim_n \sigma_n$.

Verification:

First we show that the construction is well-defined. Note that (5.21) holds for $n = 0$. For any $k \geq 0$ suppose (5.21) holds for $n = k$. Then by Lemma 5.4.2, we have $\hat{D}_k(\sigma_{k+1}) \leq 1 - 3 \cdot 2^{-k-3}$. As $\hat{M}_{k+1}(\sigma_{k+1}) \leq 2^{|\sigma_{k+1}|} \cdot M_{k+1}(\lambda) < 2^{|\sigma_{k+1}|}$, then

$$\begin{aligned}
\hat{D}_{k+1}(\sigma_{k+1}) &\leq \hat{D}_k(\sigma_{k+1}) + 2^{-|\sigma_{k+1}| - (k+1) - 2} \cdot \hat{M}_{k+1}(\sigma_{k+1}) \\
&\leq 1 - 3 \cdot 2^{-k-3} + 2^{-|\sigma_{k+1}| - (k+1) - 2} \cdot 2^{|\sigma_{k+1}|} \\
&= 1 - 2^{-k-2},
\end{aligned}$$

which shows that (5.21) also holds for $n = k + 1$. Thus, the construction is well-defined and it ensures that (5.21) holds for every $n \in \mathbb{N}$.

By (5.21), for all $i \leq n$, $2^{-|\sigma_i|-i-2} \cdot \hat{M}_i(\sigma_n) \leq \hat{D}_n(\sigma_n) \leq 1 - 2^{-n-1} < 1$. Then $\hat{M}_i(\sigma_n) \leq 2^{-|\sigma_i|-i-2}$ for all $i \leq n$. Thus, for each i it holds that $\limsup_n M_i(X \upharpoonright n) \leq 2^{-|\sigma_i|-i-2}$, i.e. $X \notin \mathsf{Succ}(M_i)$.

On the other hand, by Lemma 5.4.2 we have $\mathrm{K}_Q(\sigma_{n+1}) \leq q_n \cdot |\sigma_{n+1}|$ for all n. Then the effective Hausdorff dimension of X is no greater than 1/2. With Theorem 5.3.1 it implies that $\dim(X) = 1/2$. This completes the proof of Theorem 5.4.1. $\square$

Corollary 5.4.3. $\mathsf{Succ}[\mathit{dSLS}] \neq \mathsf{Succ}[\mathit{SLS}] = \mathsf{Succ}[\mathit{LS}]$.

5.5 Summary

We have studied the strength of monotonous strategies, which bet in a predetermined way (decidably-sided martingales). In the case of computable strategies we have seen that they are as strong as the unrestricted strategies, while in the case of mixtures of computable strategies (strongly left-c.e. supermartingales) they are significantly weaker. On the other hand, for casino sequences of effective Hausdorff dimension less than 1/2, there exists a universal strongly left-c.e. separable strategy which succeeds on all of them.

Limitations of our methods and open problems

For the proof of our main results, Theorems 5.3.2 and 5.4.1, Lemma 5.3.4 plays a very essential role. It is the main tool for us to control the potential growth of a left-c.e. supermartingale. For that, we need a canonical approximation to the supermartingale. This is the reason that our method only works for strongly left-c.e. supermartingales, not left-c.e. supermartingales in general.

We have already shown that $\mathsf{Succ}[\mathit{dSLS}] \neq \mathsf{Succ}[\mathit{LS}]$ and we also know that

$$\mathsf{Succ}[\mathit{dSLS}] \subseteq \mathsf{Succ}[\mathit{dLS}] \subseteq \mathsf{Succ}[\mathit{LS}].$$

However it is still not clear whether either of the inclusion relationship is strict. Separating the randomness notions induced by these classes of supermartingales still remain open questions.

Question 2. *If a left-c.e. supermartingale succeeds on X, does there exist a left-c.e. decidably-sided (or separable) supermartingale which succeeds on X as well?*

Question 3. *If a left-c.e. decidably-sided supermartingale succeeds on X, does there exist a strongly left-c.e. decidably-sided (or separable) supermartingale which succeeds on X as well?*

As already mentioned in § 5.4, another limitation of our methods is that it only applies to total computable prediction functions, as opposed to partial computable functions. The notion of decidably-sided supermartingale could be further generalized. We say an f-sided supermartingale is *partially decidably-sided* if f is a partial computable function. So for a partially decidably-sided supermartingale, when the decision predicate is undefined, there will be no bias presented on the two outcomes. A left-c.e. partially decidably-sided supermartingale is also called a *Kastergales*. And a strongly left-c.e. partially decidably-sided supermartingale is more like a *Hitchgale*. Both notions of supermartingales are defined to question the strength of the notion of left-c.e. supermartingales. As reported in [20] as well as in [21, §7.9], the following questions have been asked.

Question 4 (Kastermans)**.** *If a left-c.e. supermartingale succeeds on X, does there exist a Kastergale which succeeds on X as well?*

Question 5 (Hitchcock)**.** *If a left-c.e. supermartingale succeeds on X, does there exist a Hitchgale which succeeds on X as well?*

At the moment, both of them remain open. Our Theorem 5.4.1 could be seen as one step to answer these questions.

Chapter 6

Betting with Restrictions on Wagers

In a real casino, usually the wagers, i.e. the amount of money one player puts on some outcome, are restricted in many ways. For example,

(i) the wagers are always multiples of some fixed minimum wager, such as one dollar, which is a natural restriction by real money papers;

(ii) the casinos might set an upper bound for the wagers, to avoid aggressive strategies, such as the infamous roulette system "martingale" mentioned at the beginning of Chapter 5.

Restriction (i) inspired one to think about integer-valued (super)martingales, which are supermartingales that only take integer values. The randomness notion induced by computable integer-valued martingales is called integer-valued randomness, which has been investigated in the papers by Bienvenu, Stephan, and Teutsch [16], and by Herbert [27]. The computational power of integer-valued random sequences has been studied in the paper by Barmpalias, Downey, and McInerney [11]. Their work also motivated further studies on the power of restricted wager strategies, in the case where the wagers are restricted to a fixed set of arbitrary rationals, not only integers. Given a set of sequences X, an X-valued strategy is one that is restricted on wagers in X. Given two finite sets A, B of rationals, by Chalcraft, Dougherty, Freiling, and Teutsch [19], A-valued strategies can successfully replace any B-valued strategy, if and only if there exists $r \geq 0$ such that $B \subseteq r \cdot A$ (where $r \cdot A$ denotes the multiples of the elements of A with r). This characterization was extended to infinite sets, with some additional conditions, in the paper by Peretz and Bavly [41].

Remarkably, Teutsch [49] (and Peretz [40] who fixed a small flaw there) demonstrated a *savings paradox* by constructing a casino sequence on which allows integer-wager strategies to succeed, producing unbounded wealth inside the casino, but any player who attempts to save an unbounded amount by removing it from the casino, is forced to bankruptcy. In the notions of success from § 4.3,

> **savings paradox:** there is a sequence X such that there is an integer-valued martingale succeed on it, but no integer-valued supermartingale could successfully save on it.

However, in the proof of Proposition 4.3.6 we demonstrated the following

> **savings trick:** given any supermartingale M, there exists a supermartingale N computable from M such that it successfully saves on every sequence M succeeds on.

So for the class of computable supermartingales with no restriction on wagers, such a sequence can never be found.

In the purpose of analyzing the role the granularity of betting strategies plays in such dichotomy, following the line of research with these developments, we consider general classes of *granular betting strategies*. Given a nondecreasing function $g\colon \mathbb{N} \mapsto \mathbb{N}$, we call $2^{-g(s)}$ the *granule* of g at stage s. A g-granular strategy is a strategy whose wagers at strings of length s are multiples of $2^{-g(s)}$. The granule may be interpreted as the *value or purchasing power of one currency unit* at certain stage, and decrease of the granules may be interpreted as the result of inflation. As it comes up during the research, we also consider classes of *timid betting strategies*, which are subclasses of granular betting strategies, where the wager at every stage is bounded above by a constant times the granule. This actually corresponds to the restriction (ii) of a real casino.

Clearly, the absolute values of g are not important when considering classes of g-granular strategies without restriction on the amount of initial capital. And as we will see in later sections even with restriction on the amount of initial capital, the absolute values of g are not important as long as the initial capital allows one to bet at least one granule at some stage. So we only care about the rate of increase of g, or the rate of the decrease of the granule, which could also be interpreted as *inflation rate*. We will distinguish between *fine granularity* (where g is a fast order) and *coarse granularity* (where g is a constant function or slow order).

First in § 6.1 we give formal definitions of granular (super)martingales and timid (super)martingales and explore some basic properties about them.

Then in § 6.2 and § 6.3 we study how the granularity affects classes of betting strategies with different initial capitals. We show for a fine granularity g, the initial capital of a betting strategy does not really matter, as every every g-granular supermartingale is inferior to a g-granular supermartingale with arbitrary small initial capital. While for a coarse granularity g, the picture is different: on the one hand, we show that there is a g-granular martingale such that no g-granular supermartingale with less initial capital is superior to it; on the other hand, we also show that every g-granular supermartingale is inferior to a family of g-granular martingales with arbitrary small initial capital, as long as the initial capital permits betting one granule at some stage. That is to say, in terms of a single supermartingale, a coarse granularity really makes difference for the winning power of supermartingales with different initial capitals; whereas it makes no difference in terms of a class of supermartingales. This part of results are included in the paper which is still in preparation by Barmpalias and Fang [4].

Finally we study how the granularity affects the dichotomy between the "savings paradox" and the "savings trick" demonstrated at the beginning. In § 6.4 we show that in the case of a fine granularity, the "savings trick" still works. For the case of a coarse granularity g, first in § 6.5 we show that a weaker version of "savings paradox" occurs, i.e. there is a g-granular martingale M such that for any computable g-granular supermartingale N there exists a sequence X on which M succeeds but N cannot successfully saves. It is a weaker version because the sequence X witnesses the paradox is not uniform as in the "savings paradox" showed by Teutsch [49] for inter-valued martingales. Although we do not yet have a uniform solution in general, in § 6.6 we found a uniform "savings paradox" among all g-timid supermartingales. However, in § 6.7, we show that given an order g, for any g-timid supermartingale M, there is always a family $\{N_i\}$ of g-granular martingales such that there is always at least one of them successfully saves on sequences M succeeds on. It implies that if there exists a uniform "savings paradox" for a slow order g, the g-granular supermartingale diagonalizes against all g-granular saving strategies must not be a g-timid supermartingale, which is indeed the case in the "savings paradoxes" shown before. This part of results are included in the paper submitted by Barmpalias and Fang [5].

It is also worth mentioning that there is yet another motivation to study granular strategies. In the paper by Barmpalias, Lewis-Pye, and Teutsch [13], granular strategies have played a crucial role in the analysis of coding with restricted redundancy by random sequences. Roughly speaking, the coding with redundancy g we discussed extensively in Chapter 3 correspond to g-granular strategies. Hence, understanding how the granularity of a betting strategy restricts its power can be used in order to study the impact of restrictions on redundancy functions can have on a coding process.

6.1 Granularity and Timidness of Supermartingales

Generally speaking, the 'granularity' of a function $f\colon 2^{<\omega} \mapsto \mathbb{Q}$ measures how far the values of f are from being integers. While "timidness" indicates that the values of f are linearly bounded by the granules.

Definition 6.1.1. Let $g\colon \mathbb{N} \mapsto \mathbb{N}$ be a nondecreasing function.

- A function $f\colon 2^{<\omega} \mapsto \mathbb{Q}$ is *g-granular* if $f(\sigma) \cdot 2^{g(|\sigma|)} \in \mathbb{N}$ for any $\sigma \in 2^{<\omega}$, and if in addition there is a constant c such that $f(\sigma) \leq c \cdot 2^{-g(|\sigma|)}$ for all $\sigma \in 2^{<\omega}$, then it is *g-timid*.
- A (super)martingale M is *g-granular* or *g-timid* if its wager w_M is g-granular or g-timid as a function, respectively.

The class of all computable g-granular (g-timid) martingales is denoted by CM_g (CM_{g^*}) . And the class of all computable g-granular (g-timid) supermartingales is denotes by CS_g (CS_{g^*}) . We also denote CM_0 (CS_0) to be the class of CM_g (CS_g) with g being the constant function 0. Recall that integer valued martingales are computable martingales which only take values of integers. By Lemma 6.1.2 to be proved below, CM_0 is actually the same as the class of all integer valued martingales. As we will also study the interplay of granularity

CM_g	the class of all computable g-granular martingales
CM_{g^*}	the class of all computable g-timid martingales
$\mathsf{CM}_g(l)$	the class of all computable g-granular martingales with initial capital l
CS_g	the class of all computable g-granular supermartingales
CS_{g^*}	the class of all computable g-timid supermartingales
$\mathsf{CS}_g(l)$	the class of all computable g-granular Supermartingales with initial capital l
CM_0	the class of all computable integer-valued martingales
CS_0	the class of all computable integer-valued supermartingales

Table 6.1 Classes of computable strategies with granularity g

and initial capital for supermartingales, we denote the class of all g-granular computable (super)martingales with initial capital l by $\mathsf{CM}_g(l)$ ($\mathsf{CS}_g(l)$).

All the classes of g-granular (super)martingales we introduced here are summarized in Table 6.1.

Note that for a g-granular (super)martingale M if $M(\lambda) \cdot 2^{g(0)} \in \mathbb{N}$ then its cover $\widehat{M}$ is a g-granular function. Moreover, if its marginal saving M^* is also a g-granular function, then M is also a g-granular function.

Actually, we show that any g-granular (super)martingale M can be easily transformed into a (super)martingale N which is a g-granular function and its saving function takes integer values, while $|M(\sigma) - N(\sigma)| = \mathbf{O}(1)$.

Lemma 6.1.2. *Given a nondecreasing $g\colon \mathbb{N} \mapsto \mathbb{N}$ and a g-granular supermartingale M, there exists a supermartingale N such that N is a g-granular function, $S_N(\sigma) \in \mathbb{N}$ for all $\sigma \in 2^{<\omega}$ and $|M(\sigma) - N(\sigma)| = 2$. Moreover N is computable from M, g, and if M is a martingale then N is also a martingale.*

Proof. Let $\widehat{M}$ be the cover of M, and S_M be the saving. For all $\sigma \in 2^{<\omega}$, let $N(\sigma) = \widehat{M}(\sigma) - \lfloor S_M(\sigma) \rfloor + \lceil M(\lambda) \rceil - M(\lambda)$. Clearly N is a g-granular function. As $\lfloor S_M(\sigma) \rfloor - \lceil M(\lambda) \rceil + M(\lambda) \le S_M(\sigma) \le \widehat{M}(\sigma)$, by Proposition 4.1.4 N is a supermartingale. For all $\sigma \in 2^{<\omega}$, $S_N(\sigma) = \lfloor S_M(\sigma) \rfloor \in \mathbb{N}$. And $N(\sigma) - M(\sigma) = S_M(\sigma) - \lfloor S_M(\sigma) \rfloor + \lceil M(\lambda) \rceil - M(\lambda) \in [0, 2]$.

Finally, note that N is computable from M and g, and in the case when M is a martingale we have $S_M(\sigma) = 0$ for all σ, so $N = \widehat{N}$ is a martingale. □

As granularity is in conflict with scaling operations on the wagers, so the saving method in the proof of Proposition 4.3.6 breaks down in the case of granular strategies. However the following property can be salvaged, albeit non-uniformly.

Proposition 6.1.3. *Given a nondecreasing function* $g\colon \mathbb{N} \mapsto \mathbb{N}$ *and a* g*-granular supermartingale* M*, there exists a* g*-granular supermartingale* N *computable from* M *such that* $\mathsf{Succ}(M) \subseteq \mathsf{SSucc}(N)$*.*

Proof. In the case where $\lim_n M(X \upharpoonright n) = \infty$ we can simply let $N = M$. Otherwise let q be a positive rational upper bound of $r := \liminf_n M(X \upharpoonright n)$ and let N have initial capital a rational $N(\lambda) > q - r$. Let m_{-1} be such that for each $n \geq m_{-1}$ we have $M(X \upharpoonright n) \geq r$. Then let N produce part of the bets of M along a given sequence Y as follows: wait until some $n_0 \geq m_{-1}$ such that $M(Y \upharpoonright n) < q$, and then let m_0 be the least $m > n_0$ such that $M(Y \upharpoonright m) > q + 1$ (if such number does not exist, let $m_0 = \infty$). In the interval $[m_{-1}, n_0)$ the strategy N does not place any bets, while in $[n_0, m_0]$ it places the same bets that M does, along Y. Hence $N(Y \upharpoonright n) - N(Y \upharpoonright n_0) = M(Y \upharpoonright n) - M(Y \upharpoonright n_0)$, and since $r \geq M(X \upharpoonright n)$, we have $M(Y \upharpoonright n) - M(Y \upharpoonright n_0) > r - q$ for each $n \in [n_0, m_0]$. Hence $N(Y \upharpoonright n) > N(Y \upharpoonright n_0) + (r - q) > 0$ for each $n \in [n_0, m_0)$. Moreover, in the case that $m_0 < \infty$, $M(Y \upharpoonright m_0) - M(Y \upharpoonright n_0) > 1$, so $N(Y \upharpoonright m_0) > N(Y \upharpoonright n_0) + 1$. This process repeats in the same way, defining the intervals $[m_{i-1}, n_i)$ where N does not bet, and the adjacent intervals $[n_i, m_i)$ where N copies the bets of M. If for some i we have $m_i = \infty$ then after position $Y \upharpoonright n_i$ strategy N always copies the bets of M along Y.

Inductively, we can show that N is non-negative, and for each $i \geq -1$ such that $m_i < \infty$ and each $n > m_i$ we have $N(Y \upharpoonright n) > i + 1$. Moreover clearly N is g-granular and computable from M. Finally, in the case where $\limsup_n M(Y \upharpoonright n) = \infty$, the endpoints n_i, m_i are defined for all $i \in \mathbb{N}$, which means that $\lim_n N(Y \upharpoonright n) = \infty$. □

Corollary 6.1.4. *For any nondecreasing function* $g\colon \mathbb{N} \mapsto \mathbb{N}$*,*

$$\mathsf{Succ}[\mathit{CM}_g] = \mathsf{SSucc}[\mathit{CM}_g] = \mathsf{Succ}[\mathit{CS}_g] = \mathsf{SSucc}[\mathit{CS}_g].$$

Proof. By Proposition 4.3.4 and 6.1.3, and the fact that $\mathit{CM}_g \subset \mathit{CS}_g$, we have the following observation which concludes the above equations:

$$
\begin{tikzcd}
 & \mathsf{SSucc}[\mathit{CS}_g] \arrow[dr, phantom, sloped, "\subseteq"] & \\
\mathsf{SSucc}[\mathit{CM}_g] \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \mathsf{Succ}[\mathit{CS}_g] \subseteq \mathsf{SSucc}[\mathit{CM}_g]. \\
 & \mathsf{Succ}[\mathit{CM}_g] \arrow[ur, phantom, sloped, "\subseteq"] &
\end{tikzcd}
$$

□

Remember that the result by Teutsch [49] shows that $\mathsf{Succ}[\mathit{CM}_0] \neq \mathsf{Save}[\mathit{CS}_0]$, while by Corollary 4.3.11, $\mathsf{Succ}[\mathit{CM}] = \mathsf{Save}[\mathit{CS}]$. Whether $\mathsf{Save}[\mathit{CS}_g] = \mathsf{Succ}[\mathit{CS}_g]$ holds under some condition of the granularity g is one of the main questions of this chapter.

6.2 Fine Granularity and Initial Capital

For a class of (super)martingales, if there is no granularity restriction, the initial capital of a (super)martingale is actually not important, because one can easily build another (super)martingale with arbitrary positive initial capital but the same betting coefficient and then both two (super)martingales have the same set of success sequences. However, once there is a granularity restriction, this may no longer possible as larger initial capital permits the use of finer betting coefficient.

In this section, we show that for a fine granularity g, the class of g-granular (super)martingales with arbitrary small positive initial capital is still as powerful as the class of (super)martingales without any restrictions.

Theorem 6.2.1 (Barmpalias and Fang [4])**.** *Let g be a fast order. Given any $\epsilon > 0$ and any g-granular supermartingale M there exists a g-granular martingale N with initial capital ϵ such that* $\mathsf{Succ}(M) \subseteq \mathsf{Succ}(N)$*. Moreover, N is computable from M, g.*

Proof. Given any $\epsilon > 0$ and any g-granular supermartingale M. Let w_M be the wager function of M. As $\sum_{i\in\mathbb{N}} 2^{-g(i)} < \infty$, let $n_0 = \min\{n\colon \sum_{i\geq n} 2^{-g(i)} < \epsilon/2\}$ and $m = \max\{M(\sigma)\colon \sigma \in 2^{n_0-1}\}$.

Now we define a g-granular martingale N by induction. Given $\sigma \in 2^{<\omega}$, let

$$N(\sigma) = \begin{cases} \epsilon & \text{if } \sigma \in 2^{<n_0}, \\ N(\sigma^-) + \llbracket w_M(\sigma) \cdot \frac{\epsilon}{2m} \rrbracket_{g(|\sigma|)} & \text{if } \sigma \in 2^{\geq n_0}. \end{cases}$$

By definition for any $\sigma \in 2^{\geq n_0}$, we have

$$\begin{aligned} N(\sigma) &> N(\sigma^-) + w_M(\sigma) \cdot \frac{\epsilon}{2m} - 2^{-g(|\sigma|)} \\ &\geq N(\sigma \upharpoonright (n_0 - 1)) + \sum_{n_0 \leq i \leq |\sigma|} (w_M(\sigma \upharpoonright i) \cdot \frac{\epsilon}{2m} - 2^{-g(i)}) \\ &\geq \epsilon + \frac{\epsilon}{2m} \cdot (M(\sigma) - M(\sigma \upharpoonright (n_0 - 1))) - \sum_{n_0 \leq i \leq |\sigma|} 2^{-g(i)} \\ &\geq \epsilon + \frac{\epsilon}{2m} \cdot (M(\sigma) - m) - \epsilon/2 \\ &= \frac{\epsilon}{2m} \cdot M(\sigma). \end{aligned} \tag{6.1}$$

As $M(\sigma) \geq 0$, $N(\sigma) > 0$. Then it is easy to check that N is a well-defined g-granular martingale and it is computable from M, g.

Given a sequence X, for any $n \geq n_0$, by (6.1) we have

$$N(X \upharpoonright n) > \frac{\epsilon}{2m} \cdot M(X \upharpoonright n).$$

Then if $X \in \mathsf{Succ}(M)$, $X \in \mathsf{Succ}(N)$ as well. Thus, $\mathsf{Succ}(M) \subseteq \mathsf{Succ}(N)$. □

The following corollary follows directly from Theorem 6.2.1.

Corollary 6.2.2. *If g be a computable fast order, for any $k > 0$ it holds* $\mathsf{Succ}[\mathit{CS}_g(k)] = \mathsf{Succ}[\mathit{CS}]$.

6.3 Coarse Granularity and Initial Capital

In contrast to the case of fine granularities, for a coarse granularity g, there is a computable g-granular supermartingale which is not inferior to any computable g-granular supermartingales with less initial capital.

At first, we show an example for computable integer-valued supermartingales.

Theorem 6.3.1 (Barmpalias and Fang [4]). *Given $l \in \mathbb{N}^+$, there exists an integer-valued computable martingale M with initial capital $l+1$ such that for any integer-valued supermartingale N with initial capital l, $\mathsf{SSucc}(M) \setminus \mathsf{Succ}(N) \neq \emptyset$.*

Proof. The martingale M is defined as follows. It always bets 1 on side '1' if possible. Formally,

1. $M(\lambda) = l + 1$.

2. For any $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$,

$$M(\sigma ^\frown i) = \begin{cases} M(\sigma) + 1 & \text{if } M(\sigma) \geq 1 \ \& \ i = 1, \\ M(\sigma) - 1 & \text{if } M(\sigma) \geq 1 \ \& \ i = 0, \\ M(\sigma) & \text{if } M(\sigma) < 1. \end{cases}$$

Now suppose N is a integer-valued supermartingale with initial capital l and $\mathsf{SSucc}(M) \subseteq \mathsf{Succ}(N)$.

Lemma 6.3.2. *If for some $\tau \in 2^{<\omega}$ and integer $k \geq 1$ it holds that $N(\tau) < k + 1 \leq M(\tau)$, then there exists $\sigma \succeq \tau$ such that $N(\sigma) < k \leq M(\sigma)$.*

Proof. Suppose it is wrong, then for all $\sigma \succeq \tau$ if $M(\sigma) \geq k$ then $N(\sigma) \geq k$.

As $M(\tau) \geq k + 1$, we have $M(\tau ^\frown 1) \geq k + 2, M(\tau ^\frown 0) \geq k$. Then $N(\tau ^\frown 1) \geq k, N(\tau ^\frown 0) \geq k$. While $N(\tau) \leq k$, it must be the case that $N(\tau ^\frown 1) = N(\tau ^\frown 0) = k$.

Now that $M(\tau ^\frown 11) \geq k+3, M(\tau ^\frown 10) \geq k+1$, then $N(\tau ^\frown 11) \geq k, N(\tau ^\frown 10) \geq k$. By $N(\tau ^\frown 1) = k$, we have $N(\tau ^\frown 11) = N(\tau ^\frown 10) = k$.

In the same way, by induction we have $N(\tau ^\frown 1^i) = k$ for all $i \in \mathbb{N}$. Then $\tau ^\frown 1^\omega \notin \mathsf{Succ}(N)$. While $M(\tau ^\frown 1^i) \geq k + i + 1$ for all $i \in \mathbb{N}$. Thus, $\tau ^\frown 1^\omega \in \mathsf{SSucc}(M)$, which contradicts the assumption that $\mathsf{SSucc}(M) \subseteq \mathsf{Succ}(N)$. □

As $N(\lambda) < l + 1 \leq M(\lambda)$, by Lemma 6.3.2 there exists $\sigma_1 \in 2^{<\omega}$ such that $N(\sigma_1) < l \leq M(\sigma_1)$. By iteratively applying Lemma 6.3.2, we get $\sigma_1 \preceq \sigma_2 \preceq \cdots \preceq \sigma_l$ such that $N(\sigma_l) < 1 \leq M(\sigma_l)$. That is to say $N(\sigma_l) = 0$ and $M(\sigma_l) \geq 1$. Then $N(\sigma_l ^\frown 1^i) = 0$ for all $i \in \mathbb{N}$ and $\sigma_l ^\frown 1^\omega \notin \mathsf{Succ}(N)$. While $M(\sigma_l ^\frown 1^i) \geq i + 1$ for all $i \in \mathbb{N}$ and $\sigma_l ^\frown 1^\omega \in \mathsf{SSucc}(M)$. This contradicts the assumption that $\mathsf{SSucc}(M) \subseteq \mathsf{Succ}(N)$. Thus, $\mathsf{SSucc}(M) \setminus \mathsf{Succ}(N) \neq \emptyset$. □

Now we show that the above theorem holds for any coarse granularity as well. The proof idea is essentially the same, but now we need to pay attention to the granularity. We will construct the martingale M and a witness X for each g-granular supermartingale N directly.

Theorem 6.3.3 (Barmpalias and Fang [4]). *Let g be a constant function or slow order. For any $l \in \mathbb{N}^+$ there exists a g-granular martingale M computable from g with initial capital l such that for any g-granular supermartingale N with initial capital less than l, $\mathsf{SSucc}(M) \setminus \mathsf{Succ}(N) \neq \emptyset$.*

Proof. The g-granular martingale M is defined as follows. It always bets one granule on side '1' if possible. Formally,

1. $M(\lambda) = l$.
2. For any $\sigma \in 2^{<\omega}$ and $i \in \{0, 1\}$,

$$M(\sigma ^\frown i) = \begin{cases} M(\sigma) + 2^{-g(|\sigma|+1)} & \text{if } M(\sigma) \geq 2^{-g(|\sigma|)} \ \& \ i = 1, \\ M(\sigma) - 2^{-g(|\sigma|+1)} & \text{if } M(\sigma) \geq 2^{-g(|\sigma|)} \ \& \ i = 0, \\ M(\sigma) & \text{if } M(\sigma) < 2^{-g(|\sigma|)}. \end{cases}$$

Note that the way we define M makes it a g-granular function computable from g. So in case $M(\sigma) < 2^{-g(|\sigma|)}$, actually $M(\sigma) = 0$.

Now suppose N is a g-granular supermartingale with initial capital less than l. Define a sequence X as follows,

$$X(n) = \begin{cases} 1 & \text{if } N(X \upharpoonright n ^\frown 0) \geq N(X \upharpoonright n ^\frown 1), \\ 0 & \text{if } N(X \upharpoonright n ^\frown 0) < N(X \upharpoonright n ^\frown 1). \end{cases}$$

Fix some $n \in \mathbb{N}$. Note that $N(X \upharpoonright n ^\frown 0) + N(X \upharpoonright n ^\frown 1) \leq 2N(X \upharpoonright n)$.
In case $X(n) = 1$: By definition, $N(X \upharpoonright n ^\frown 1) - N(X \upharpoonright n ^\frown 0) \leq 0$. Then

$$N(X \upharpoonright n + 1) = N(X \upharpoonright n ^\frown 1) \leq N(X \upharpoonright n).$$

And

$$M(X \upharpoonright n + 1) = \begin{cases} M(X \upharpoonright n) + 2^{-g(n+1)} & \text{if } M(X \upharpoonright n) > 0 & (6.2) \\ M(X \upharpoonright n) & \text{if } M(X \upharpoonright n) = 0 & (6.3) \end{cases}$$

In case $X(n) = 0$: By definition, $N(X \upharpoonright n^\frown 0) - N(X \upharpoonright n^\frown 1) < 0$. As N is a g-granular supermartingale, $w_N(X \upharpoonright n^\frown 0) = \frac{N(X\upharpoonright n^\frown 0) - N(X\upharpoonright n^\frown 1)}{2}$ is an integer multiple of $2^{-g(n+1)}$. Then $N(X \upharpoonright n^\frown 0) - N(X \upharpoonright n^\frown 1) \leq -2 \cdot 2^{-g(n+1)}$. Thus,

$$N(X \upharpoonright n+1) = N(X \upharpoonright n^\frown 0) \leq N(X \upharpoonright n) - 2^{-g(n+1)}.$$

And

$$M(X \upharpoonright n+1) = \begin{cases} M(X \upharpoonright n) - 2^{-g(n+1)} & \text{if } M(X \upharpoonright n) > 0 \quad (6.4) \\ M(X \upharpoonright n) & \text{if } M(X \upharpoonright n) = 0 \quad (6.5) \end{cases}$$

In any case $N(X \upharpoonright n+1) \leq N(X \upharpoonright n)$ for all n. So $\lim_{n\to\infty} N(X \upharpoonright n) \leq N(\lambda) < l$ and $X \notin \mathsf{Succ}(N)$.

And in any case $M(X \upharpoonright n+1) - M(X \upharpoonright n) \geq N(X \upharpoonright n+1) - N(X \upharpoonright n)$ for all n. So $M(X \upharpoonright n) \geq N(X \upharpoonright n) - N(\lambda) + M(\lambda) > 0$, which means only (6.2) and (6.4) apply. Therefore,

$$\begin{aligned} M(X \upharpoonright n) &= M(\lambda) + \sum_{0\leq i<n \& X(i)=1} 2^{-g(i+1)} - \sum_{0\leq i<n \& X(i)=0} 2^{-g(i+1)} \\ &= l + \sum_{0<i\leq n} 2^{-g(i)} - 2 \sum_{0\leq i<n \& X(i)=0} 2^{-g(i+1)} \\ &\geq l + \sum_{0\leq i\leq n} 2^{-g(i)} + 2 \sum_{0\leq i<n} (N(X \upharpoonright i+1) - N(X \upharpoonright i)) \\ &\geq l + \sum_{0\leq i\leq n} 2^{-g(i)} - 2N(\lambda) \\ &\geq \sum_{0\leq i\leq n} 2^{-g(i)} - l. \end{aligned}$$

As $\sum_{i\in\mathbb{N}} 2^{-g(i)} = \infty$, we have $\lim_{n\to\infty} M(X \upharpoonright n) = \infty$. So $X \in \mathsf{SSucc}(M)$. Thus, $\mathsf{SSucc}(M) \setminus \mathsf{Succ}(N) \neq \emptyset$. $\square$

From Theorem 6.3.3 one is tempted to guess that $\mathsf{Succ}[\mathit{CS}_g(k)] \subsetneq \mathsf{Succ}[\mathit{CS}_g(l)]$ if $k < l$, given g as a coarse granularity. However, this turns out to be wrong.

Theorem 6.3.4 (Barmpalias and Fang [4])**.** *Let g be a constant function or slow order. For any $k \in \mathbb{N}$ and any g-granular supermartingale M with initial capital $(k+1) \cdot 2^{-g(0)}$, there is a countable class $\mathcal{C}$ of g-granular martingales with initial capital $k \cdot 2^{-g(0)}$ such that $\mathsf{Succ}(M) \subseteq \mathsf{Succ}[\mathcal{C}]$. Moreover, all the martingales in $\mathcal{C}$ are computable from M, g.*

Proof. Given any supermartingale M as stated in the theorem, let w be the wager function of M. Inductively, we define a martingale T, which alway bet all its capital on the side where M prefers.

- $T(\lambda) = k \cdot 2^{-g(0)}$.
- For $\tau \in 2^{>0}$,

$$T(\tau) = \begin{cases} 2 \cdot T(\tau^-) & \text{if } w(\tau) \geq 0, \\ 0 & \text{if } w(\tau) < 0. \end{cases}$$

And for each $\sigma \in 2^{<\omega}$, we construct a martingale N_σ. It always bets all its capital along σ and then at strings extending σ it just bets the same wager as M as long as it is possible, while for all other strings, it does not put any bet. The formal definition is by induction as follows.

- $N_\sigma(\lambda) = k \cdot 2^{-g(0)}$.
- For $\tau \in 2^{>0}$,

$$N_\sigma(\tau) = \begin{cases} 0 & \text{if } \tau \not\preceq \sigma \wedge \sigma \not\prec \tau, \\ 2 \cdot N_\sigma(\tau^-) & \text{if } \tau \preceq \sigma, \\ N_\sigma(\tau^-) + \min\{|w(\tau)|, N_\sigma(\tau^-)\} & \text{if } \tau \succ \sigma \wedge w(\tau) \geq 0, \\ N_\sigma(\tau^-) - \min\{|w(\tau)|, N_\sigma(\tau^-)\} & \text{if } \tau \succ \sigma \wedge w(\tau) < 0. \end{cases}$$

It is easy to check that T and all $\{N_\sigma\}_{\sigma \in 2^{<\omega}}$ are well-defined g-granular martingales of initial capital $k \cdot 2^{-g(0)}$ and they are all computable from M, g. We define $\mathcal{C}$ to be $\{T\} \cup \bigcup_{\sigma \in 2^{<\omega}} \{N_\sigma\}$.

Lemma 6.3.5. *For each $\sigma \in 2^{<\omega}$, the following are true for the martingale N_σ.*

(i) $N_\sigma(\sigma \upharpoonright n) \geq k \cdot 2^{n-g(0)}$ for all $0 \leq n \leq |\sigma|$.

(ii) If $M(\sigma) \leq k \cdot 2^{|\sigma|-g(0)}$, then $N_\sigma(\tau) \geq M(\tau)$ for all $\tau \succeq \sigma$.

Proof. (i): We prove by induction on n. First, for $n = 0$, by definition we have $N_\sigma(\sigma \upharpoonright 0) = N_\sigma(\lambda) = k \cdot 2^{-g(0)}$. Now suppose $N_\sigma(\sigma \upharpoonright i) \geq k \cdot 2^{i-g(0)}$ for some $0 \leq i < |\sigma|$. Again by definition we have $N_\sigma(\sigma \upharpoonright (i+1)) = 2 \cdot N_\sigma(\sigma \upharpoonright i) = k \cdot 2^{i+1-g(0)}$.

(ii): We prove by induction on τ. First, for $\tau = \sigma$, as $M(\sigma) \leq k \cdot 2^{|\sigma|-g(0)}$, by (i) we have $N_\sigma(\sigma) \geq k \cdot 2^{|\sigma|-g(0)} \geq M(\sigma)$. Now suppose $N_\sigma(\tau) \geq M(\tau)$ for some $\tau \succeq \sigma$. Then for any $i \in \{0,1\}$, we have $|w(\tau^\frown i)| \leq M(\tau) \leq N_\sigma(\tau)$. Thus, by definition $N_\sigma(\tau^\frown i) = N_\sigma(\tau) + w(\tau^\frown i) \geq M(\tau) + w(\tau^\frown i) = M(\tau^\frown i)$. $\square$

Lemma 6.3.6. *For each $X \in \mathsf{Succ}(M)$, if there exists some $n \in \mathbb{N}$ such that $M(X \upharpoonright n) \leq k \cdot 2^{n-g(0)}$ then $X \in \mathsf{Succ}(N_{X\upharpoonright n})$.*

Proof. As $M(X \upharpoonright n) \leq k \cdot 2^{n-g(0)}$, then by Lemma 6.3.5 $N_{X\upharpoonright n}(\tau) \geq M(\tau)$ for all $\tau \succeq X \upharpoonright n$. Especially, $N_{X\upharpoonright n}(X \upharpoonright i) \geq M(X \upharpoonright i)$ for all $i \geq n$. As $X \in \mathsf{Succ}(M)$, i.e. $\limsup_{i\to\infty} M(X \upharpoonright i) = \infty$, then $\limsup_{i\to\infty} N_{X\upharpoonright n}(X \upharpoonright i) = \infty$ as well. Thus, $X \in \mathsf{Succ}(N_{X\upharpoonright n})$. $\square$

Lemma 6.3.7. *For each $X \in \mathsf{Succ}(M)$, if $M(X \upharpoonright n) > k \cdot 2^{n-g(0)}$ for all $n \in \mathbb{N}$, then $X \in \mathsf{Succ}(T)$.*

Proof. Given $X \in \mathsf{Succ}(M)$ such that $M(X \upharpoonright n) > k \cdot 2^{n-g(0)}$ for all $n \in \mathbb{N}$, for all $i > 0$, let

$$c_i = 1 + \frac{w(X \upharpoonright i)}{M(X \upharpoonright (i-1))}.$$

Then for all $n > 0$, we have

$$M(X \upharpoonright n) \leq M(\lambda) \cdot \prod_{0<i\leq n} c_i.$$

As $M(\lambda) = (k+1) \cdot 2^{-g(0)}$, then

$$(k+1) \cdot 2^{-g(0)} \cdot \prod_{0<i\leq n} c_i > k \cdot 2^{n-g(0)},$$

i.e.

$$\prod_{0<i\leq n} c_i > \frac{k}{k+1} \cdot 2^n \geq 2^{n-1}.$$

On the other hand, clearly, for all $i > 0$, $c_i \leq 2$. Suppose $w(X \upharpoonright n) < 0$ for some n, then $c_n < 1$,

$$\prod_{0<i\leq n} c_i \leq \prod_{0<i\leq n-1} c_i \leq 2^{n-1},$$

which is a contradiction. Thus, $w(X \upharpoonright n) \geq 0$ for all $n > 0$. Then by definition, $T(X \upharpoonright n) = 2^n \cdot T(\lambda) = k \cdot 2^{n-g(0)}$, i.e. $X \in \mathsf{Succ}(T)$. $\square$

Lemma 6.3.6 and Lemma 6.3.7 together show that $\mathsf{Succ}(M) \subseteq \mathsf{Succ}[\mathcal{C}]$. $\square$

Corollary 6.3.8. *Let g be a constant function or computable slow order. For any two numbers $k < l$ such that $k \geq 2^{-g(n)}$ for some $n \in \mathbb{N}$, it holds* $\mathsf{Succ}[\mathit{CM}_g(k)] = \mathsf{Succ}[\mathit{CS}_g(l)]$.

Proof. Given g and k, l be as stated in the theorem, obviously $\mathsf{Succ}[\mathit{CM}_g(k)] \subseteq \mathsf{Succ}[\mathit{CS}_g(l)]$. Let $n_0 = \min\{i \colon k \geq 2^{-g(i)}\}$ and $d = \min\{i \in \mathbb{N} \colon i \geq l \cdot 2^{n_0+g(n_0)}\}$. We define a function $h \colon \mathbb{N} \mapsto \mathbb{N}$ as $h(n) = g(n+n_0)$ for all $n \in \mathbb{N}$. Clearly, h is a nondecreasing function. By Theorem 6.3.4, $\mathsf{Succ}[\mathit{CM}_h(k \cdot 2^{-h(0)})] = \mathsf{Succ}[\mathit{CS}_h((k+1) \cdot 2^{-h(0)})]$ holds for all $k \in \mathbb{N}^+$. By iterating we obtain $\mathsf{Succ}[\mathit{CM}_h(2^{-h(0)})] = \mathsf{Succ}[\mathit{CS}_h(d \cdot 2^{-h(0)})]$.

Given any $X \in \mathsf{Succ}[\mathit{CS}_g(l)]$, let M be a supermartingale in $\mathit{CS}_g(l)$ such that $X \in \mathsf{Succ}(M)$. Then we have $\limsup_{n\to\infty} M(X \upharpoonright n) = \infty$ and $M(X \upharpoonright n_0) \leq l \cdot 2^{n_0} \leq d \cdot 2^{-h(0)}$. Thus, $X \upharpoonright [n_0, \infty) \in \mathsf{Succ}[\mathit{CS}_h(M(X \upharpoonright n_0))] \subseteq \mathsf{Succ}[\mathit{CS}_h(d \cdot 2^{-h(0)})] = \mathsf{Succ}[\mathit{CM}_h(2^{-h(0)})]$. Let N be a martingale in $\mathit{CM}_h(2^{-h(0)})$ such that $X \upharpoonright [n_0, \infty) \in \mathsf{Succ}(N)$. Now define an another martingale N' as follows. For $\sigma \in 2^{<\omega}$,

$$N'(\sigma) = \begin{cases} 2^{-g(n_0)} & \text{if } |\sigma| \leq n_0, \\ N(\sigma \upharpoonright [n_0, |\sigma|)) & \text{if } |\sigma| > n_0. \end{cases}$$

Clearly, N' is a well-defined martingale in $\mathit{CM}_g(2^{-g(n_0)})$. Moreover, we have $X \in \mathsf{Succ}(N')$. Thus, $X \in \mathsf{Succ}[\mathit{CM}_g(2^{-g(n_0)})] \subseteq \mathsf{Succ}[\mathit{CM}_g(k)]$. Hence, $\mathsf{Succ}[\mathit{CM}_g(k)] = \mathsf{Succ}[\mathit{CS}_g(l)]$ □

6.4 Fine Granularity and Saving Strategies

Now we study the interact of granularities and notions of success.

Theorem 6.4.1 (Barmpalias and Fang [5])**.** *Let g be a fast order. Given any supermartingale M, there exists a g-granular supermartingale N computable from M, g such that* $\mathsf{Succ}(M) \subseteq \mathsf{Save}(N)$.

Given a supermartingale M, we will construct N directly. N chooses some large enough initial capital and then bets according to M, by letting its wager for the next stage be the wager for next stage of M multiplied with the ratio between the current capital of N and M. Whenever the capital of M becomes doubled or larger compared to its previous capital marker, N consumes 1 and sets the current capital of M to be a new capital marker. Then N uses the rest of its capital to do the betting according to M. However, there might be some issue caused by the restriction of granularity. To solve this, N chooses its wager to be the value which satisfies the granularity requirement and has the largest absolute value less than or equal to the absolute value of the wager for next stage of M multiplied with the ratio between the current available capital of N and M. By this means, we can prove that the ratio between the current available capital of N and M is bounded from below if the initial ratio is large enough. And on the other hand, N will successfully save wherever M succeeds.

Proof. Let G be some integer such that $G > 2 + \sum_{n \in \mathbb{N}} 2^{-g(n)}$. Without loss of generality, we assume $M(\sigma) \geq 1$ for all $\sigma \in 2^{<\omega}$. Because otherwise, we could define $M' = M + 1$ and use M' instead of M in the following argument.

Let w_M be the wager of M. For any $\sigma \in 2^{<\omega}$, let $I(\sigma)$ be the set of all numbers n_i such that

1. $n_0 = 0$,
2. $n_i < n_{i+1} \leq |\sigma|$ and n_{i+1} is the least number such that $M(\sigma \upharpoonright n_{i+1}) \geq 2M(\sigma \upharpoonright n_i)$.

We let $l(\sigma) = |I(\sigma)| - 1$.

Now we define three g-granular functions w, N' and N by induction.

1. $w(\lambda) = 0,\ N'(\lambda) = N(\lambda) = \lfloor G \cdot M(\lambda) \rfloor + 1$.
2. For any $\sigma \succ \lambda$, let

$$\begin{aligned} w(\sigma) &= \left[\!\!\left[w_M(\sigma) \cdot \frac{N'(\sigma^-) - l(\sigma^-)}{M(\sigma^-)} \right]\!\!\right]_{g(|\sigma|)}, \\ N'(\sigma) &= N'(\sigma^-) + w(\sigma), \\ N(\sigma) &= N'(\sigma) - l(\sigma^-). \end{aligned}$$

Clearly that they are g-granular functions and N is computable from M, g.

Lemma 6.4.2. *For all $\sigma \in 2^{<\omega}$, $N'(\sigma) - l(\sigma) > M(\sigma)$.*

Proof. For any $\sigma \succ \lambda$, by definition we have

$$w(\sigma) \geq w_M(\sigma) \cdot \frac{N'(\sigma^-) - l(\sigma^-)}{M(\sigma^-)} - 2^{-g(|\sigma|)}.$$

Then

$$\begin{aligned}\frac{N'(\sigma)-l(\sigma^-)}{M(\sigma)} &\geq \frac{N'(\sigma^-)+w_M(\sigma)\cdot\frac{N'(\sigma^-)-l(\sigma^-)}{M(\sigma^-)}-2^{-g(|\sigma|)}-l(\sigma^-)}{M(\sigma^-)+w_M(\sigma)}\\ &= \frac{N'(\sigma^-)-l(\sigma^-)}{M(\sigma^-)}-\frac{2^{-g(|\sigma|)}}{M(\sigma)}\\ &\geq \frac{N'(\sigma^-)-l(\sigma^-)}{M(\sigma^-)}-2^{-g(|\sigma|)}.\end{aligned}$$

If $I(\sigma)=I(\sigma^-)$, then we have $l(\sigma)=l(\sigma^-)$ and

$$\frac{N'(\sigma)-l(\sigma)}{M(\sigma)} \geq \frac{N'(\sigma^-)-l(\sigma^-)}{M(\sigma^-)}-2^{-g(|\sigma|)}.$$

If $I(\sigma)\neq I(\sigma^-)$, then we have $|\sigma|\in I(\sigma)$, $l(\sigma)=l(\sigma^-)+1$ and $M(\sigma)\geq 2^{l(\sigma)}M(\lambda)$. Then

$$\begin{aligned}\frac{N'(\sigma)-l(\sigma)}{M(\sigma)} &= \frac{N'(\sigma)-l(\sigma^-)}{M(\sigma)}-\frac{1}{M(\sigma)}\\ &\geq \frac{N'(\sigma^-)-l(\sigma^-)}{M(\sigma^-)}-2^{-g(|\sigma|)}-2^{-l(\sigma)}.\end{aligned}$$

Thus,

$$\begin{aligned}\frac{N'(\sigma)-l(\sigma)}{M(\sigma)} &\geq \frac{N'(\lambda)-l(\lambda)}{M(\lambda)}-\sum_{n=1}^{|\sigma|}2^{-g(n)}-\sum_{n=1}^{l(\sigma)}2^{-n}\\ &\geq G-\sum_{n\in\mathbb{N}}2^{-g(n)}-1\\ &> 1,\end{aligned}$$

which implies $N'(\sigma)-l(\sigma)>M(\sigma)$. □

Lemma 6.4.3. *N is a g-granular supermartingale and its saving function is $N'-N$.*

Proof. By Lemma 6.4.2, for any $\sigma\in 2^{<\omega}$, $N'(\sigma)>l(\sigma)>0$. And for $i\in\{0,1\}$, as $l(\sigma\frown i)\geq l(\sigma)$, we also have

$$N(\sigma\frown i)=N'(\sigma\frown i)-l(\sigma)\geq N'(\sigma\frown i)-l(\sigma\frown i)>0.$$

On the other hand, by definition we have for all $\sigma \in 2^{<\omega}$ and $i \in \{0,1\}$,

$$\begin{aligned} N'(\sigma^\frown i) - N(\sigma^\frown i) &= l(\sigma) \le l(\sigma^\frown i), \\ N(\sigma^\frown 0) + N(\sigma^\frown 1) &= 2(N'(\sigma) - l(\sigma)) \le 2N(\sigma), \\ N(\sigma^\frown i) - N(\sigma^\frown (1-i)) &= 2w(\sigma^\frown i), \\ N'(\sigma) &= N(\lambda) + \sum_{\tau \preceq \sigma} w(\tau). \end{aligned} \tag{6.6}$$

Thus, by definition N is a supermartingale, w is its wager and N' is its cover. By Proposition 4.1.3 its saving function $S_N = N' - N$.

As w is a g-granular function, N is a g-granular supermartingale. □

Given any sequence $X \in \mathsf{Succ}(M)$, we have $\limsup_{n\to\infty} M(X \upharpoonright n) = \infty$. Then the size of the set $I(X \upharpoonright n)$ is unbounded, i.e. $\lim_{n\to\infty} l(X \upharpoonright n) = \infty$. By (6.6), we have

$$\lim_{n\to\infty} S_N(X \upharpoonright n) = \lim_{n\to\infty} l(X \upharpoonright (n-1)) = \infty.$$

Thus, $X \in \mathsf{Save}(N)$.

Therefore, $\mathsf{Succ}(M) \subseteq \mathsf{Save}(N)$. □

Together with Corollary 6.2.2, Theorem 6.4.1 has the following consequence, which indicates that in the case of fine granularity, any class of computable fine granular (super)martingales is interchangeable with the class of computable (super)martingales, and within them the three notions of success are also interchangeable.

Corollary 6.4.4. *For any computable fast order g,*

$$\mathsf{Save}[\mathit{CS}_g] = \mathsf{Succ}[\mathit{CM}_g] = \mathsf{SSucc}[\mathit{CM}_g] = \mathsf{Succ}[\mathit{CS}_g] = \mathsf{SSucc}[\mathit{CS}_g] = \mathsf{Succ}[\mathit{CS}].$$

6.5 Coarse Granularity and Saving Strategies

However, as in the interact of granularities and initial capitals, for the case of coarse granularity, a difference for the notions of success appears, and the situation here seems more chaotic.

Theorem 6.5.1 (Barmpalias and Fang [5]). *Let g be a constant function or slow order. There exists a g-granular martingale M computable from g such that for any g-granular supermartingale T it holds that $\mathsf{Succ}(M) \setminus \mathsf{Save}(T) \neq \emptyset$.*

Proof. The g-granular martingale M is defined the same as in the proof of Theorem 6.3.3. Suppose T is an arbitrary g-granular supermartingale. By Lemma 6.1.2, we can assume T is a g-granular function. Let w_T and $\widehat{T}$ be its wager and cover, respectively.

Now we construct a sequence $X \in \mathsf{Succ}(M) \setminus \mathsf{Save}(T)$. The construction is done by induction.

$\boxed{\textbf{Construction of } X\textbf{:}}$

For $n \geq 0$, suppose $X \upharpoonright n$ has already been defined. To simplify the expressions we let

$$\begin{aligned} m(n) &= M(X \upharpoonright n), \\ t(n) &= T(X \upharpoonright n), \\ w(n) &= w_T(X \upharpoonright n ^\frown 1), \\ q(n) &= \lfloor t(n)/m(n) \rfloor. \end{aligned}$$

Then we define

$$X(n) = \begin{cases} 1 & \text{if } w(n) \leq q(n) \cdot 2^{-g(n+1)} \\ 0 & \text{if } w(n) > q(n) \cdot 2^{-g(n+1)} \end{cases}.$$

$\boxed{\textbf{Explanation:}}$

Before we come to the verification part, it might be better to explain the intuition behind the construction. Let S_T be the saving function of T and $r(n) = t(n) - q(n) \cdot m(n)$ for all $n \geq 0$. The function $r(n)$ is called the *remainder* at *level* $q(n)$. We define a function L to keep record of the total loss of the remainder.

- $L(0) = 0$.
- $L(n+1) = L(n) + r(n) + (q(n) - q(n+1)) \cdot m(n+1) - r(n+1)$.

The initial capital of T might be much larger than the initial capital of M, in which case we say that T has a large advantage over M. Our idea is to construct the sequence X such that along it that advantage will be restrained in terms of the quotient between the value of T and M.

However, the advantage is not measured by the exact quotient, but the integer part of the quotient $q(n)$ and the remainder $r(n)$. Roughly speaking, $q(n)$ determines the advantage level, while $r(n)$ is the advantage at that level.

With this intuition in mind, $w(n) > q(n) \cdot 2^{-g(n+1)}$ means that T attempts to jump to a higher level. In that case we let X go to the other direction to prevent this. Then T will suffer from this by losing some of its advantage, either a loss of advantage at the same level or decreasing its level. And $w(n) \leq q(n) \cdot 2^{-g(n+1)}$ means that T wants to keep its level or jump to lower level. In that case we let it go, and T's advantage will be restrained. Moreover, in any case, if T makes any savings, i.e. S_T increases, it will directly result in an equal amount of loss of its advantage.

Whenever T's advantage at level k is used up, either by making a wrong attempt to change level or by consumption, its advantage level will then decrease to $k-1$ or lower. Note that the maximal advantage of T at level $q(n)$ is determined by the capital of M at the stage when T enters that level. We call it the capacity of the level $q(n)$. By all this setting, we achieves that T's total savings is upper bounded by the total loss of advantages at all levels L. While L is upper bounded by the sum of the capacities from level $q(0)$ down to some level $\leq q(0)$ in which it finally ends up. As it is a finite amount, T cannot archive infinite savings, i.e. S_T reaches a finite limit.

On the other hand, M continually gains capital along X except when T tries to jump to higher level. While in that case, compared to otherwise smoothly gain capital, M's loss is at most 2 times the loss of T's advantage. As T's total loss L is bounded, M will eventually gain infinite capital.

Verification:

The formal verification is done by the following lemmas.

Lemma 6.5.2. *For all $n \geq 0$, $m(n) \geq 2^{-g(n+1)}$.*

Proof. As M and T are g-granular functions, we have m and t are g-granular functions and $m(n) \geq 0, t(n) \geq 0$ for all $n \geq 0$.

At first by definition $m(0) = M(\lambda) = 1 \geq 2^{-g(0)}$. Suppose k is the least number such that $m(k+1) < 2^{-g(k+2)}$, then we must have $m(k+1) = 0$. Also as $m(k) \geq 2^{-g(k+1)}$, it must be the case that $g(k) = g(k+1)$, $m(k) = 2^{-g(k+1)}$ and $w(k) > q(k) \cdot 2^{-g(k+1)}$. On the other hand, as $m(k) = 2^{-g(k+1)} = 2^{-g(k)}$ and

t is g-granular, we have $t(k) = q(k) \cdot m(k) = q(k) \cdot 2^{-g(k+1)}$. Thus, $w(k) > t(k)$, which is a contradiction. □

Lemma 6.5.2 ensures that the function $q(n)$ is well defined.

Lemma 6.5.3. *For all $n \geq 0$, $q(n+1) \leq q(n)$ and if $q(n+1) = q(n)$ then $r(n+1) \leq r(n)$.*

Proof. Fix any $n \geq 0$.

In case $X(n) = 1$: we have

$$\begin{aligned} w(n) &\leq q(n) \cdot 2^{-g(n+1)} \\ m(n+1) &= m(n) + 2^{-g(n+1)} \\ t(n+1) &\leq t(n) + w(n) \end{aligned}$$

As $q(n) = \lfloor t(n)/m(n) \rfloor \leq t(n)/m(n)$, then

$$\begin{aligned} q(n+1) &= \left\lfloor \frac{t(n+1)}{m(n+1)} \right\rfloor \\ &\leq \left\lfloor \frac{t(n) + \frac{t(n)}{m(n)} 2^{-g(n+1)}}{m(n) + 2^{-g(n+1)}} \right\rfloor \\ &= \lfloor t(n)/m(n) \rfloor \\ &= q(n). \end{aligned}$$

If $q(n+1) = q(n)$, then

$$\begin{aligned} r(n+1) &= t(n+1) - q(n) \cdot m(n+1) \\ &\leq t(n) + q(n) \cdot 2^{-g(n+1)} - q(n) \cdot (m(n) + 2^{-g(n+1)}) \\ &= t(n) - q(n) \cdot m(n) \\ &= r(n). \end{aligned}$$

In case $X(n) = 0$: we have

$$\begin{aligned} w(n) &> q(n) \cdot 2^{-g(n+1)} \\ m(n+1) &= m(n) - 2^{-g(n+1)} \\ t(n+1) &\leq t(n) - w(n). \end{aligned}$$

Since w is g-granular, we have $w(n) \geq (q(n)+1) \cdot 2^{-g(n+1)}$. As $q(n)+1 = \lfloor t(n)/m(n) \rfloor + 1 > t(n)/m(n)$, then

$$\begin{aligned} q(n+1) &= \left\lfloor \frac{t(n+1)}{m(n+1)} \right\rfloor \\ &\leq \left\lfloor \frac{t(n) - \frac{t(n)}{m(n)} 2^{-g(n+1)}}{m(n) - 2^{-g(n+1)}} \right\rfloor \\ &= \lfloor t(n)/m(n) \rfloor \\ &= q(n) \end{aligned}$$

If $q(n+1) = q(n)$, then

$$\begin{aligned} r(n+1) &= t(n+1) - q(n) \cdot m(n+1) \\ &\leq t(n) - (q(n)+1) \cdot 2^{-g(n+1)} - q(n) \cdot (m(n) - 2^{-g(n+1)}) \\ &= t(n) - q(n) \cdot m(n) - 2^{-g(n+1)} \\ &= r(n) - 2^{-g(n+1)}. \end{aligned} \tag{6.7}$$

□

From Lemma 6.5.3 we know that $\{q(n)\}$ is a nonincreasing nonnegative integer sequence. Then it must have a limit. And the value of $q(n)$ changes only finitely many times. Let $n_0 < n_1 < \cdots < n_d$ be all the positions where $q(n+1) < q(n)$ and we denote $I = \{n_0, n_1, \ldots, n_d\}$.

Lemma 6.5.4. *L is nondecreasing and if $X(n) = 0$, then $L(n+1) - L(n) \geq 2^{-g(n+1)}$. Moreover, $\lim_{n\to\infty} L(n) < \infty$.*

Proof. By the definition of L, we have

$$L(n+1) - L(n) = r(n) + (q(n) - q(n+1)) \cdot m(n+1) - r(n+1).$$

By Lemma 6.5.3, in case $q(n+1) = q(n)$, we have $r(n) - r(n+1) \geq 0$. Then $L(n+1) - L(n) \geq 0$. If $X(n) = 0$, by (6.7) we have $L(n+1) - L(n) \geq 2^{-g(n+1)}$. In case $q(n+1) < q(n)$, as $r(n+1) \leq m(n+1) - 2^{-g(n+1)}$, we also have $L(n+1) - L(n) \geq 2^{-g(n+1)}$. So L is nondecreasing and if $X(n) = 0$, then $L(n+1) - L(n) \geq 2^{-g(n+1)}$.

On the other hand,

$$\begin{aligned} L(n) &= L(0) + r(0) + \sum_{0<i\le n} (q(i-1) - q(i)) \cdot m(i) - r(n) \\ &\le \sum_{i\in I\cap(0,n]} (q(i-1) - q(i)) \cdot m(i). \end{aligned}$$

Thus,

$$\lim_{n\to\infty} L(n) \le \sum_{i\in I} (q(i-1) - q(i)) \cdot m(i) < \infty.$$

The last inequality holds because I is a finite set. □

Lemma 6.5.5. *For all $n \ge 0$, $S_T(n) \le L(n)$.*

Proof. At first we have $S_T(0) = L(0) = 0$. Now fix any $n \ge 0$.

In case $X(n) = 1$:

$$\begin{aligned} S_T(n+1) - S_T(n) &= w(n) + t(n) - t(n+1) \\ &\le q(n) \cdot 2^{-g(n+1)} + q(n) \cdot (m(n+1) - 2^{-g(n+1)}) + r(n) \\ &\qquad - q(n+1) \cdot m(n+1) - r(n+1) \\ &= r(n) + (q(n) - q(n+1)) \cdot m(n+1) - r(n+1) \\ &= L(n+1) - L(n). \end{aligned}$$

In case $X(n) = 0$:

$$\begin{aligned} S_T(n+1) - S_T(n) &= -w(n) + t(n) - t(n+1) \\ &\le -q(n) \cdot 2^{-g(n+1)} + q(n) \cdot (m(n+1) + 2^{-g(n+1)}) + r(n) \\ &\qquad - q(n+1) \cdot m(n+1) - r(n+1) \\ &= r(n) + (q(n) - q(n+1)) \cdot m(n+1) - r(n+1) \\ &= L(n+1) - L(n). \end{aligned}$$

So $S_T(n+1) - S_T(n) \le L(n+1) - L(n)$. By induction, we have $S_T(n) \le L(n)$ for all $n \ge 0$. □

Lemma 6.5.5 and Lemma 6.5.4 together show that S_T reaches a finite limit on X, i.e. $X \notin \mathsf{Save}(T)$.

Lemma 6.5.6. *For any n, $m(n) \geq \sum_{0 \leq i \leq n} 2^{-g(i)} - 2L(n)$.*

Proof. By the definition of m and Lemma 6.5.4, we have

$$\begin{aligned} m(n) &= 2^{-g(0)} + \sum_{0<i\leq n \& X(i-1)=1} 2^{-g(i)} - \sum_{0<i\leq n \& X(i-1)=0} 2^{-g(i)} \\ &= \sum_{0\leq i\leq n} 2^{-g(i)} - 2 \sum_{0<i\leq n \& X(i-1)=0} 2^{-g(i)} \\ &\geq \sum_{0\leq i\leq n} 2^{-g(i)} - 2 \sum_{0<i\leq n \& X(i-1)=0} (L(i) - L(i-1)) \\ &\geq \sum_{0\leq i\leq n} 2^{-g(i)} - 2L(n). \end{aligned}$$

□

Lemma 6.5.6 and Lemma 6.5.4 together show that $X \in \mathsf{Succ}(M)$, which completes our proof. □

Corollary 6.5.7. *Let g be a constant function or slow order. There exists a g-granular martingale M compuatble from g such that for any class of g-granular supermartingales $\mathcal{C}$ with $\sum_{S\in\mathcal{C}} S(\lambda) < \infty$ it holds that $\mathsf{Succ}(M) \setminus \mathsf{Save}[\mathcal{C}] \neq \emptyset$.*

Proof. The martingale M is defined the same as in the proof of Theorem 6.5.1. Given a class of g-granular supermartingales $\mathcal{C}$ with $\sum_{S\in\mathcal{C}} S(\lambda) < \infty$, we define a supermartingale T by summing up all supermartingales in $\mathcal{C}$, i.e. $T(\sigma) = \sum_{S\in\mathcal{C}} S(\sigma)$. Note that T is a g-granular martingale, and $\mathsf{Save}[\mathcal{C}] \subseteq \mathsf{Save}(T)$.

Now we apply Theorem 6.5.1 to the supermartingale T, and get a sequence $X \in \mathsf{Succ}(M) \setminus \mathsf{Save}(T) \subseteq \mathsf{Succ}(M) \setminus \mathsf{Save}[\mathcal{C}]$. □

Note that the sum of the initial capitals of all computable g-granular supermartingales does not converge, and our proof of Theorem 6.5.1 lacks uniformity. It is still a question that whether $\mathsf{Succ}[\mathit{CM}_g]$ equals $\mathsf{Save}[\mathit{CS}_g]$ for a computable slow order g.

6.6 Timid Supermartingales of Coarse Granularity

We notice that the martingales we constructed in the proofs of Theorems 6.3.1, 6.3.3 and 6.5.1 are actually g-timid martingale. Our following result shows that within the class of g-timid strategies of coarse granularity, the "savings paradox" appears.

Theorem 6.6.1 (Barmpalias and Fang [5])**.** *Let g be a constant function or slow order. There exists a g-timid martingale M computable from g such that for any countable class $\mathcal{C}$ of g-timid supermartingales* $\mathsf{Succ}(M) \setminus \mathsf{Save}[\mathcal{C}] \neq \emptyset$.

Proof. The g-timid martingale M is defined the same as in the proof of Theorem 6.3.3. Let $\{T_i\}$ be a list of all the g-timid supermartingales in $\mathcal{C}$. And let $\{c_i\}$ be a sequence of positive integers such that for each i, the wager of T_i on strings of length k is upper bounded by $(c_i - 1) \cdot 2^{-g(k)}$.

In the same way as in previous section, at stage $n \geq 0$, assuming $X \upharpoonright n$ has already been defined, we will determine $X(n)$. We let

$$\begin{aligned} m(n) &= M(X \upharpoonright n), \\ t_i(n) &= T_i(X \upharpoonright n), \\ w_i(n) &= w_{T_i}(X \upharpoonright n^\frown 1), \\ \ell(n) &= \max_{i \leq n} m(i). \end{aligned}$$

We will view every stage as a process which consists of a saving step where strategies $\{T_i\}$ make savings and a betting step where we choose the next bit of X. In order to avoid overloaded notation, in the following arguments at a specific step of a stage when n is clear from the context, we use m, t_i, w_i, ℓ and m_i', t_i', w_i', ℓ' to refer the values of each corresponding functions at the beginning of the step and at the end of the step (which is also the value at the beginning of the next step), respectively.

Now let $m_0 = m$, $q_0 = \lfloor t_0/m_0 \rfloor$ and $r_0 = t_0 - q_0 \cdot m_0$. For each $i > 0$, assuming that m_{i-1}, r_{i-1} are defined, define inductively

$$\begin{aligned} m_i &= m_{i-1} - r_{i-1}, \\ q_i &= \lfloor (t_i + c_i \cdot r_{i-1})/m_i \rfloor, \\ r_i &= t_i + c_i \cdot r_{i-1} - q_i \cdot m_i. \end{aligned}$$

We will ensure that $m > 0$ at all steps of all states of the process. Then $m_i > r_i \geq 0$ for all i, and all the divisions are well defined. m_i', q_i', r_i' will be used in the same fashion as t_i'. Let g^*, g^+ denote the *current granule* and the *next* *granule*, respectively. We say that t_i *requires attention* at some state if $\ell \geq i+1$ and $w_i \neq q_i \cdot g^+$.

Construction of X:

At the betting step of stage n let $i \leq \ell$ be the least such that t_i requires attention. If no such i exists, let $X(n) = 1$. Otherwise, if $w_i < q_i \cdot g^+$ let $X(n) = 1$ and if $w_i > q_i \cdot g^+$ let $X(n) = 0$.

Verification:

Instead of making a detailed calculation as in the verification of Theorem 6.5.1, now we will verify it in a more general way. For this nested construction we have the following observation.

Lemma 6.6.2. *At the betting step of stage n, if none of $t_i, i \leq k$ requires attention but $X(n) = 0$ is chosen, then $m_k - r_k > g^+$.*

Proof. Given the situation as stated, suppose $m_k - r_k = g^+$. Then $m_j = g^+$ and $r_j = 0$ for all $j > k$. Hence $w_j \leq t_j \leq t_j + c_j \cdot r_{j-1} = q_j \cdot g^+$ for all $j > k$. This means that in this case $X(n) = 1$ will be chosen, which is a contradiction. □

Lemma 6.6.2 actually ensures that $m > 0$ at all stages, thereby $m_i > r_i \geq 0$ for all i and all the divisions are well defined.

Lemma 6.6.3. *At each step of stage n, for any $k \leq \ell$, if $q'_i = q_i$ for all $i < k$, then*

i) $q'_k \leq q_k$*; and if* $q'_k = q_k$ *then* $r'_k \leq r_k$*;*

ii) for a saving step, if $q'_k = q_k$, then the marginal saving of t_k is no more than $r_k - r'_k$;

iii) for a betting step, if $q'_k = q_k$ and for some $i \leq k$, t_i requires attention, then $r'_j \leq r_j - g^+$ for all $i \leq j \leq k$.

Proof. For $k = 0$, just notice that at stages where t_0 does not require attention, we have $t'_0 = t_0 \pm w_0 = q_0 \cdot m_0 + r_0 \pm q_0 \cdot g^+ = q_0 \cdot m'_0 + r_0$. With the observation from Lemma 6.6.2 we have $q'_0 \leq q_0$ and if $q'_0 = q_0$ then $r'_0 \leq r_0$. The rest of the proof then follows directly from the verification in the proof of Theorem 6.5.1, as the outcome of X will be chosen in the same way.

Now inductively let $d > 0$ and assume that Lemma 6.6.3 holds for all $k < d$. We prove Lemma 6.6.3 for $k = d$. By applying Lemma 6.6.3 to all $k < d$, we get $r'_i \leq r_i$ for all $i < k$.

At a saving step, $m'_k = m_k$. Then

$$\begin{aligned} t_k - t'_k &= (r_k + q_k \cdot m_k - c_k \cdot r_{k-1}) - (r'_k + q'_k \cdot m'_k - c_k \cdot r'_{k-1}) \\ &= r_k - r'_k - c_k \cdot (r_{k-1} - r'_{k-1}) \leq r_k - r'_k, \end{aligned}$$

$$q'_k \cdot m'_k + r'_k = t'_k + c_k \cdot r'_{k-1} \leq t_k + c_k \cdot r_{k-1} \leq q_k \cdot m'_k + r_k.$$

Thus, i) and ii) hold.

At a betting step, depending on the outcome chosen, we have

$$m'_k = \begin{cases} m_k + g^+ + \sum_{0 \leq i < k}(r_i - r'_i) \geq m_k + g^+ & \text{if } X(n) = 1, \\ m_k - g^+ + \sum_{0 \leq i < k}(r_i - r'_i) \geq m_k - g^+ & \text{if } X(n) = 0, \end{cases}$$

$$t'_k + c_k \cdot r'_{k-1} \leq \begin{cases} t_k + w_k + c_k \cdot r_{k-1} = q_k \cdot m_k + r_k + w_k & \text{if } X(n) = 1, \\ t_k - w_k + c_k \cdot r_{k-1} = q_k \cdot m_k + r_k - w_k & \text{if } X(n) = 0. \end{cases}$$

We distinguish among the following three cases.

Case a): *none of $t_i, i \leq k$ requires attention.*

In this case $w_k = q_k \cdot g^+$. Then

$$q'_k \cdot m'_k + r'_k = t'_k + c_k \cdot r'_{k-1} \leq q_k \cdot m'_k + r_k.$$

Thus, with Lemma 6.6.2 we have $q'_k \leq q_k$ and if $q'_k = q_k$ then $r'_k \leq r_k$.

Case b): *t_k requires attention.*

In this case

$$\begin{cases} w_k \leq (q_k - 1) \cdot g^+ & \text{if } X(n) = 1, \\ w_k \geq (q_k + 1) \cdot g^+ & \text{if } X(n) = 0, \end{cases}$$

Then

$$q'_k \cdot m'_k + r'_k = t'_k + c_k \cdot r'_{k-1} \leq q_k \cdot m'_k + r_k - g^+.$$

Thus, we have $q'_k \leq q_k$ and if $q'_k = q_k$ then $r'_k \leq r_k - g^+$.

Case c): *t_i requires attention for some $i < k$.*

By induction hypothesis, we already have $r'_j \leq r_j - g^+$ for all $i \leq j < k$. On the other hand,

$$m'_k \geq m_k - g^+ + \sum_{0 \leq i < k} (r_i - r'_i) \geq m_k,$$

$$\begin{aligned} q'_k \cdot m'_k + r'_k &= t'_k + c_k \cdot r'_{k-1} \leq t_k + (c_k - 1) \cdot g^+ + c_k \cdot (r_{k-1} - g^+) \\ &= t_k + c_k \cdot r_{k-1} - g^+ = q_k \cdot m_k + r_k - g^+ \leq q_k \cdot m'_k + r_k - g^+. \end{aligned}$$

Thus, we have $q'_k \leq q_k$ and if $q'_k = q_k$ then $r'_k \leq r_k - g^+$.

This completes the proof of Lemma 6.6.3 □

Lemma 6.6.4. *For each i, there is a stage after which $\ell \geq i$ and q_j is a constant for all $j < i$.*

Proof. Inductively for any $i \geq 0$ assume that there is a stage s_0 after which $\ell \geq i$ and q_j is a constant for all $j < i$ (for $i = 0$ this is trivial). At stages after s_0, supposing we always have $\ell < i+1$, then t_j does not require attention for all $j \geq i$. So m decreases only at betting steps where for some $j < i$, t_j requires attention and the outcome 0 is chosen. By Lemma 6.6.3, in that case m decreases by g^+ and r_j decreases by at least g^+. Moreover, r_j is nonincreasing after stage s_0 for all $j < i$. So the total decrease of m after stage s_0 is at most $\sum_{j<i} r_i[s_0]$. Thus, for any $s \geq s_0$, $M(X \upharpoonright s) \geq \sum_{s \geq s_0} 2^{-g(s+1)} - \sum_{j<i} r_i[s_0]$. Then there must be some $s_1 \geq s_0$ such that $M(X \upharpoonright s_1) \geq i+1$, i.e. $\ell \geq i+1$. Then by Lemma 6.6.3 along X after stage s_1, q_i is a series of nonincreasing positive integers. Thus, there is a stage later than s_1 after which q_i is a constant as well. □

Lemma 6.6.5. $\limsup_n M(X \upharpoonright n) = \infty$ *and for each i, T_i has only finite savings.*

Proof. For each i, by Lemma 6.6.4 there is a stage s_i after which $\ell \geq i$ and q_j is a constant for all $j \leq i$. Clearly, we have $\ell \to \infty$ and then $\limsup_n M(X \upharpoonright n) = \infty$. Fix i, by Lemma 6.6.3, after stage s_i, r_i is nonincreasing and at each saving step T_i saves no more than $r_i - r'_i$. It follows that after stage s_i strategy T_i can save at most $r_i[s_i]$. Thus, its total savings is finite. □

Lemma 6.6.5 shows that $X \in \mathsf{Succ}(M) \setminus \mathsf{Save}[\mathcal{C}]$, which completes our proof.

□

Corollary 6.6.6. *For any constant function or computable slow order g,*

$$\mathsf{Save}[CS_{g^*}] \subsetneq \mathsf{Succ}[CS_{g^*}].$$

6.7 Weakness of Timid Supermartingales

We have already noticed that, in the proofs of Theorems 6.3.3, 6.5.1 and 6.6.1, we used the same g-timid martingale, which always bets one granule on '1'. Compared with Theorem 6.6.1, our last theorem reveals that for a slow order g, the class of g-timid supermartingales are not strong enough to devise a saving strategy.

Theorem 6.7.1 (Barmpalias and Fang [5])**.** *Let g be an order. For any g-timid supermartingale M, there is a countable class $\mathcal{C}$ of g-granular supermartingales such that $\mathsf{Succ}(M) \subseteq \mathsf{Save}[\mathcal{C}]$. Moreover, all the martingales in $\mathcal{C}$ are computable from M, g.*

As g is an order, given a g-timid supermartingale M, there exists a non-increasing $h\colon \mathbb{N} \mapsto \mathbb{Q}$ which tends to 0 and such that $w_\sigma \leq h(|\sigma|)$ for each σ. We will construct g-granular supermartingales T and $\{N_\rho\}_{\rho\in I}$, where I is a certain set of binary strings that will be defined below. Our idea is to use T as the main supermartingale, and $\{N_\rho\}_{\rho\in I}$ as backup supermartingales. The first backup supermartingale is N_λ. For a single backup supermartingale N_ρ, it might be closed at some stage σ, which means we will not let it bet at any string extending σ. But when this happens, we will issue another backup supermartingale N_σ which will serve as the active backup supermartingale on strings extending σ until it might be closed at some further stages. Thus, we ensure that at every stage there is exactly one backup supermartingale active. Then for the construction at every stage we only need to specify the wager of the current active backup strategy, while for all other backup strategies their wagers are 0. If N_ρ is the active backup strategy at stage σ, we define the index of the stage σ as $i_\sigma = \rho$. Our construction will ensure that $i_\sigma \preceq i_\tau$ when $\sigma \preceq \tau$. For simplicity we often omit the subscript ρ, when it is clear which backup strategy is the active one for the current stage.

During the construction, at the beginning, the stage will be considered as neutral. At neutral stages both T and the active backup supermartingale N_ρ follows M identically, in the sense of having the same wager for each bit, until some condition is met, then we will start a cycle. During cycles, both T and some backup supermartingale N_ρ attempt to gain extra capital and save later. They follow different policies. T doubles the wager of M on the same outcome. N_ρ multiplies the wager of M by a large coefficient, say b, and put it on the opposite outcome. We use the difference $r_\sigma = r(\sigma) = T(\sigma) - M(\sigma)$ to monitor the progress in the cycle. Let c_σ denote the value of r_σ at the beginning of a cycle and only update it at the end of each cycle. We end a cycle at η when r_η escapes the interval $(c_\eta/2, c_\eta + 1)$ and let the following stage be neutral until a new cycle starts.

In case $r_\eta \geq c_\eta + 1$, it means that during the cycle T gains extra 1, then we let T save 1 at the end of this cycle. Note that during such a cycle N_ρ loses $(r_\eta - c(\eta)) \cdot b$, which might be quite a lot. To make sure N_ρ is a valid supermartingale, we need to pay attention to the coefficient b. On the other hand, as N_ρ is "heavily damaged" in this cycle, we then close it by setting $N_\rho(\tau) = N_\rho(\eta)$ for all $\tau \succ \eta$, and initiate a new backup strategy N_η which receives an initial capital of $M(\eta)$ and only starts to bet at η.

In case $r_\eta \leq c_\eta/2$, it means that during this cycle T loses $c_\eta - r(\eta)$, but N_ρ gains $(c_\eta - r(\eta)) \cdot b$. We will set b appropriately such that this value is more than 1. Then we let N_ρ save 1 at the end of this cycle. While N_ρ continues to serve as the active backup supermartingale, as $N_\rho \geq M$ still holds after the saving.

By the g-timidness of M, we will ensure that once r_σ escaped from $(c_\sigma/2, c_\sigma + 1)$, it is still within $(c_\sigma/4, c_\sigma + 2)$, so that there is always space for starting a new cycle.

We will ensure that for any path X if $\limsup_n M(X \upharpoonright n) = \infty$ then one of the following will happen.

1. infinitely many backup supermartingales initiated and closed along X: in this case T successfully saves;

2. there is a last supermartingale N_ρ initiated along X, which is never closed and generates infinitely many cycles along X, each of them ending in failure for T: in this case N_ρ successfully saves.

Proof of Theorem 6.7.1. Given M as in the statement of Theorem 6.7.1, we define the saving supermartingale T, the set I of strings/stages where new cycles are initiated, and the family $\{N_\rho\}_{\rho \in I}$ of backup supermartingales. By the above discussion, for the construction at every stage we only need to specify the wa-

w_σ	wager of M at σ
v_σ	wager of T at σ
u_σ	wager of active N at σ
r_σ	$T(\sigma) - M(\sigma)$
c_σ	marker of r at the starting stage of a cycle
i_σ	index ρ of backup strategy N_ρ active at σ
s_t, s_n	saving functions for T, N respectively
h	function with $w_\sigma \le h(\|\sigma\|)$, $\lim_n h(n) = 0$

Table 6.2 Parameters for the proof of Theorem 6.7.1.

ger of the current active backup strategy. For the set i, we only need to specify i_σ at every stage σ, because then $I = \{i_\rho \mid i_\rho \neq i_{\rho^-}\}$.

Stages will be inductively classified as *neutral stages* or *cycle stages*. The last stage of a cycle is also called as (cycle) *ending stage*. A *cycle interval* is an interval from the starting stage of a cycle to an ending stage without neutral stage in between. Our strategies only save on the ending stages. So we divide each ending stage σ^- into two steps, the betting and the saving step. The value of the parameter r at the end of the betting step is denoted by r^0_σ, while at the end of saving step is denoted by r_σ.

For simplicity we often omit the subscript ρ, when it is clear which backup strategy is the active one for the current stage. We now inductively define the index i_σ of current active backup supermartingale N, the wagers v_σ, u_σ of the supermartingale T and N along with their saving functions s_t, s_n and the type of the stage σ. Remember that by our notation, the values are calculated as follows.

$$\begin{aligned} N(\sigma) &= N(\sigma^-) + u_\sigma - s_n(\sigma^-), \\ T(\sigma) &= T(\sigma^-) + v_\sigma - s_t(\sigma^-), \\ r^0_\sigma &= T(\sigma^-) + v_\sigma - M(\sigma), \\ r_\sigma &= T(\sigma) - M(\sigma). \end{aligned}$$

The parameters involved here are summarized in Table 6.2.

$\boxed{\textbf{Construction:}}$

Let $T(\lambda) = M(\lambda) + 1$, $N_\lambda(\lambda) = M(\lambda)$. Let $c_\lambda = r_\lambda = 1, i_\lambda = \lambda$, and λ be neutral.

Given $\sigma \neq \lambda$, inductively assume that $c_{\sigma^-}, i_{\sigma^-}$ and the type of σ^- have been defined, consider the cases:

1. **if σ^- is a neutral stage:** Let $v_\sigma = u_\sigma = w_\sigma$, $c_\sigma = c_{\sigma^-}$, $i_\sigma = i_{\sigma^-}$. Then
   - if $N(\sigma) > 2\lceil 2/c_\sigma \rceil$ and $h(|\sigma|) < \min\{1, c_\sigma/4\}$, let σ start an i_σ-cycle;
   - otherwise, let σ be a neutral stage.
2. **if σ^- is a cycle stage:** Let $v_\sigma = 2w_\sigma$, $u_\sigma = -\lceil 2/c_{\sigma^-} \rceil \cdot w_\sigma$. Then
   - if $r^0_\sigma \in (c_{\sigma^-}/2, c_{\sigma^-} + 1)$, let $c_\sigma = c_{\sigma^-}$, $i_\sigma = i_{\sigma^-}$ and σ be a cycle stage;
   - if $r^0_\sigma \leq c_{\sigma^-}/2$, mark σ^- as a T-*failed ending stage* and let $s_n(\sigma^-) = 1$, $c_\sigma = r_\sigma$, $i_\sigma = i_{\sigma^-}$, and σ be a neutral stage;
   - if $r^0_\sigma \geq c_{\sigma^-} + 1$, mark σ^- as a T-*successful ending stage* and let $s_t(\sigma^-) = 1$, $c_\sigma = r_\sigma$, $i_\sigma = \sigma$, and σ be a neutral stage.

$\boxed{\textbf{Verification:}}$

Lemma 6.7.2. *For any $\sigma \neq \lambda$ such that σ^- is a neutral stage, we have $N(\sigma) - M(\sigma) = N(\sigma^-) - M(\sigma^-)$ and $r_\sigma = r_{\sigma^-} = c_{\sigma^-} = c_\sigma$.*

This lemma is a trivial observation of the construction.

Lemma 6.7.3. *For any $\sigma \neq \lambda$ such that σ^- is a cycle stage, assuming η is the starting stage of that cycle interval,*

1. *if σ^- is not an ending stage, then $r_\sigma > r_\eta/2$ and $N(\sigma) > \lceil 2/r_\eta \rceil$;*
2. *if σ^- is a T-failed ending stage, then $r_\sigma > r_\eta/4$, N saves 1 and $N(\sigma) - M(\sigma) \geq N(\eta) - M(\eta)$;*
3. *if σ^- is a T-successful ending stage, then $r_\sigma \geq r_\eta$, $N(\sigma) > 0$ and T saves 1.*

Proof. First we observe that c_τ does not change at a cycle stage τ. By the construction, for all $\eta \prec \tau \preceq \sigma$, we have $v_\tau = 2w_\tau$ and $u_\tau = -\lceil 2/c_{\tau^-} \rceil \cdot w_\tau =$

$-\lceil 2/c_\eta \rceil \cdot w_\tau$. then

$$r^0_\sigma - r_\eta = \left(T(\eta) + \sum_{\eta \prec \tau \preceq \sigma} v_\tau \right) - \left(M(\eta) + \sum_{\eta \prec \tau \preceq \sigma} w_\tau \right) - (T(\eta) - M(\eta))$$
$$= \sum_{\eta \prec \tau \preceq \sigma} w_\tau = M(\sigma) - M(\eta),$$
$$N^0(\sigma) - N(\eta) = \sum_{\eta \prec \tau \preceq \sigma} u_\tau = -\left\lceil \frac{2}{c_\eta} \right\rceil \cdot \sum_{\eta \prec \tau \preceq \sigma} w_\tau = -\left\lceil \frac{2}{c_\eta} \right\rceil \cdot (r^0_\sigma - r_\eta).$$

On the other hand, by Lemma 6.7.2 $c_\eta = r_\eta$ and by construction

$$N(\eta) > 2\lceil 2/r_\eta \rceil, \quad h(|\eta|) < \min\{1, r_\eta/4\}.$$

If σ^- is not an ending stage, then

$$r_\sigma > c_{\sigma^-}/2 = c_\eta/2 = r_\eta/2, \quad r_\sigma < c_{\sigma^-} + 1 = c_\eta + 1 = r_\eta + 1,$$
$$N(\sigma) = -\lceil 2/c_\eta \rceil \cdot (r_\sigma - r_\eta) + N(\eta) > -\lceil 2/r_\eta \rceil + 2\lceil 2/r_\eta \rceil = \lceil 2/r_\eta \rceil$$

If σ^- is an ending stage, note that $|r^0_\sigma - r_{\sigma^-}| = |w_{\sigma^-}| < h(|\sigma^-|) \le h(|\eta|)$ and $r_{\sigma^-} \in (r_\eta/2, r_\eta + 1)$.

In case σ^- is a T-failed ending stage, N saves 1 and $r_\sigma = r^0_\sigma$. Then

$$r_\sigma \le c_{\sigma^-}/2 = r_\eta/2, \quad r_\sigma \ge r_{\sigma^-} - |r^0_\sigma - r_{\sigma^-}| > r_\eta/2 - h(|\eta|) > r_\eta/4,$$

and

$$N(\sigma) - N(\eta) = -\lceil 2/c_\eta \rceil \cdot (r_\sigma - r_\eta) - 1 \ge \lceil 2/c_\eta \rceil \cdot r_\eta/2 - 1 \ge 0$$
$$M(\sigma) - M(\eta) = r_\sigma - r_\eta \le r_\eta/2 - r_\eta < 0.$$

Thus, $N(\sigma) - M(\sigma) \ge N(\eta) - M(\eta)$.

In case σ^- is a T-successful ending stage, T saves 1, $r_\sigma = r^0_\sigma - 1$ and

$$r^0_\sigma \ge c_{\sigma^-} + 1 = r_\eta + 1, \quad r^0_\sigma \le r_{\sigma^-} + |r^0_\sigma - r_{\sigma^-}| < r_\eta + 1 + h(|\eta|) < r_\eta + 2,$$

Then $r_\sigma \ge r_\eta$ and

$$N(\sigma) = -\lceil 2/c_\eta \rceil \cdot (r^0_\sigma - r_\eta) + N(\eta) > -2\lceil 2/c_\eta \rceil + 2\lceil 2/c_\eta \rceil = 0.$$

□

Lemma 6.7.4. *T and all $\{N_\rho\}_{\rho \in I}$ are g-granular supermartingales.*

Proof. As the wagers of T and N_ρ are always integer multiples of the wagers of M, the g-granularity of T, N_ρ follows from the g-granularity of M. Then by construction, we only need to verify that T and N_ρ are always non-negative.

As $r_\lambda = 1 > 0$, by Lemma 6.7.2 and Lemma 6.7.3 inductively we easily get $r_\sigma > 0$ for all σ. Then $T(\sigma) = M(\sigma) + r_\sigma > 0$ for all σ, as required.

Fix $\rho \in I$, for simplicity we drop the subscript ρ for N_ρ for the rest of this proof. First for all $\sigma \not\succeq \rho$, $N(\sigma) = N(\rho) = M(\rho) \geq 0$. And by Lemma 6.7.3 $N(\sigma) > 0$ for all σ such that σ^- is a cycle stage but not T-failed ending stage. Moreover, if σ^- is a T-successful ending stage, then on all strings extending σ N takes the same value as $N(\sigma)$, which is positive. On the other hand, as ρ is a neutral stage for N and $N(\rho) - M(\rho) = 0$, by Lemma 6.7.2 and Lemma 6.7.3 inductively we get that for all σ such that σ^- is a neutral stage or T-failed ending stage, $N(\sigma) - M(\sigma) \geq 0$, i.e. $N(\sigma) \geq M(\sigma) \geq 0$. □

Lemma 6.7.5. *For any X such that $\limsup_n M(X \upharpoonright n) = \infty$, there are infinitely many cycles starting and ending along X.*

Proof. If only finitely many cycles occur along X, one of the following must hold:

(i) almost all prefixes of X are neutral;

(ii) there exists a cycle along X which never ends.

It remains to show each of the above clauses implies $\limsup_n M(X \upharpoonright n) < \infty$.

First assume that (i) holds and that η is the least prefix of X such that all prefixes of X after η are neutral. If ρ is the index of η, then for all $\eta \preceq \sigma \prec X$ we have $i_\sigma = \rho$ and $c_\sigma = c_\eta$. As $h \to 0$, then there is $\eta \preceq \tau \prec X$ such that $h(|\tau|) < \min\{1, c_\eta/4\} = \min\{1, c_\tau/4\}$. Moreover, for all $\tau \preceq \sigma \prec X$, $h(|\sigma|) < \min\{1, c_\sigma/4\}$. As no cycle starts at any prefix of X after η, then for all $\tau \preceq \sigma \prec X$, $N(\sigma) \leq 2\lceil 2/c_\sigma \rceil = 2\lceil 2/c_\eta \rceil$. As showed in the proof of Lemma 6.7.4, at a neutral stage σ it holds $N(\sigma) \geq M(\sigma)$. Hence M is bounded above along X, as required.

Second, assume that (ii) holds and at $\rho \prec X$ a cycle starts, which never ends along X. Then for all $\eta \prec X$ after ρ we have $r_\eta - r_\rho = M(\eta) - M(\rho)$. By condition for ending a cycle in the construction, it follows that r_η remains bounded above by $r_\rho + 1$ along X, hence M is bounded above along X, as required. ☐

Lemma 6.7.6. *For any X such that $\limsup_n M(X \upharpoonright n) = \infty$, either T successfully saves along X or N_ρ successfully saves along X for some $\rho \in I$.*

Proof. By the assumption on X and Lemma 6.7.5 there are infinitely many cycles along X. Since a new cycle only starts after the previous one ends, and since each ending stage is either T-successful or in T-failed, it follows that one of the following holds:

(a) there are infinitely many T-successful cycles along X;

(b) all but finitely many cycles along X are T-failed.

If (a) holds, then by Lemma 6.7.3, T successfully saves along X. If (b) holds, the indices of the initial segments of X reach a limit ρ. Hence starting from ρ and along X, the backup supermartingale N_ρ will remain active, and there will be infinitely many ρ-cycles and all of them will end in T-failure. By Lemma 6.7.3 N_ρ successfully saves along X. ☐

Let $\mathcal{C} = \{T\} \cup \bigcup_{\sigma \in I} \{N_\sigma\}$, by Lemma 6.7.6 we have $\mathsf{Succ}(M) \subseteq \mathsf{Save}[\mathcal{C}]$. Clearly, every supermartingale in $\mathcal{C}$ is computable from M, g. This completes the proof of Theorem 6.7.1. ☐

Corollary 6.7.7. *For any computable order g,*

$$\mathsf{Succ}[\mathsf{CS}_{g^*}] \subseteq \mathsf{Save}[\mathsf{CS}_g].$$

6.8 Summary

Liquidity in betting situations, in the sense of infinite divisibility of the capital, allows for certain flexibilities in the strategies, including avoiding bankruptcy while placing infinitely many bets, and saving an unbounded capital on the condition that the betting strategy is successful. Such properties are based on the fact that liquidity allows arbitrary *scaling* of the strategy, i.e. the implementation of essentially the original strategy but with arbitrarily small available capital. We have already seen that in a casino where a fixed betting unit is set, things may change, like there exists a casino sequence along which it is possible to reach unbounded profit but it is not possible to save unbounded profit by constantly withdraw the profit into some frozen account. While for a casino without fixed betting unit, such a casino sequence never exists. We study here to what extent such liquidity is necessary to lead different properties.

Image a casino with fixed betting units, but the betting units are shrinking over stages. We study whether the following two properties for classes of betting strategies with different shrinking rates.

(a) One can win with arbitrary small initial capital along any sequence that a given strategy wins;

(b) One can win by saving unbounded profit, i.e. withdrawing the profit into some frozen account, along any sequence that a given strategy reaches unbounded profit.

We show that in case the shrinking is fast, say with a rate faster than $1/n$, then such a casino shares both properties with a casino without fixed betting units. However in case the shrinking is slower, on the one hand, either property is completely possessed by such a casino. On the other hand, a weaker version (a') is still preserved.

(a') A countable family of strategies with arbitrary small initial capital can win along any sequence that a given strategy wins.

However, whether an analog for property (b) can be found is still an open problem:

Question 6. *For a computable slow order g, given a g-granular supermartingale is there always a countable family of g-granular supermartingales such that there is always one of them successfully saves on any sequence that the given supermartingale succeeds on?*

In order to know more about this question, we extended our research to the casinos where a maximal of units is exposed to the wagers. In such a casino, only "timid" strategies are allow, while "bold" strategies are forbidden. We found that for such a casino, Question 6 has a negative answer. In the meanwhile, we also show that for Question 6, if the given supermartingale is "timid", then the required countable family of g-granular supermartingales indeed can be always found. These might be some steps towards an answer for Question 6, though the question in general remains open.

References

[1] Klaus Ambos-Spies. On the strongly bounded Turing degrees of the computably enumerable sets. In *Computability and Complexity*, pages 563–598. Springer, 2017.

[2] Klaus Ambos-Spies, Decheng Ding, Yun Fan, and Wolfgang Merkle. Maximal pairs of computably enumerable sets in the computably Lipschitz degrees. *Theory of Computing Systems*, 52(1):2–27, 2013.

[3] Klaus Ambos-Spies, Nadine Losert, and Martin Monath. Array noncomputable left-c.e. reals. In preparation.

[4] George Barmpalias and Nan Fang. The importance of initial capital in betting. In preparation.

[5] George Barmpalias and Nan Fang. Granularity of wagers in games and the (im) possibility of savings. Submitted.

[6] George Barmpalias and Andrew E. M. Lewis. A c.e. real that cannot be sw-computed by any Omega number. *Notre Dame Journal of Formal Logic*, 47(2):197–209, 2006.

[7] George Barmpalias and Andrew Lewis-Pye. Computing halting probabilities from other halting probabilities. *Theoretical Computer Science*, 660: 16–22, 2017.

[8] George Barmpalias and Andrew Lewis-Pye. Limits of the Kucera-Gacs coding method. In *Post-proceedings volume of SEALS 2016 (South Eastern Logic Symposium)*. World Scientific, 2017.

[9] George Barmpalias and Andrew Lewis-Pye. Optimal redundancy in computations from random oracles. *Journal of Computer and System Sciences*, 92:1–8, 2018.

[10] George Barmpalias, Rodney G. Downey, and Noam Greenberg. Working with strong reducibilities above totally ω-c.e. and array computable degrees. *Transactions of the American Mathematical Society*, 362(2):777–813, 2010.

[11] George Barmpalias, Rodney G. Downey, and Michael McInerney. Integer-valued betting strategies and Turing degrees. *Journal of Computer and System Sciences*, 81:1387–1412, 2015.

[12] George Barmpalias, Nan Fang, and Andrew Lewis-Pye. Optimal asymptotic bounds on the oracle use in computations from Chaitin's Omega. *Journal of Computer and System Sciences*, 82(8):1283–1299, 2016.

[13] George Barmpalias, Andrew Lewis-Pye, and Jason Teutsch. Lower bounds on the redundancy in computations from random oracles via betting strategies with restricted wagers. *Information and Computation*, 251:287–300, 2016.

[14] George Barmpalias, Nan Fang, and Andrew Lewis-Pye. Monotonous betting strategies in warped casinos. *Information and Computation*, Accepted.

[15] Laurent Bienvenu, Glenn Shafer, and Alexander Shen. On the history of martingales in the study of randomness. *Electronic Journal for History of Probability and Statistics*, 5(1), 2009.

[16] Laurent Bienvenu, Frank Stephan, and Jason Teutsch. How powerful are integer-valued martingales? In *Programs, Proofs, Processes, 6th Conference on Computability in Europe, CiE, 2010, Ponta Delgada, Azores, Portugal, June 30 - July 4, 2010, Proceedings*, pages 59–68, 2010.

[17] Francesca Zaffora Blando. *From von Mises' Impossibility of a Gambling System to Probabilistic Martingales.* MSc dissertation, Institute of Logic, Language and Computation, University of Amsterdam, The Netherlands, 2015.

[18] Gregory J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22:329–340, 1975. ISSN 0004-5411.

[19] Adam Chalcraft, Randall Dougherty, Chris Freiling, and Jason Teutsch. How to build a probability-free casino. *Information and Computation*, 211: 160–164, 2012.

[20] Rodney G. Downey. Randomness, computation and mathematics. In *How the World Computes: Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings*, CiE'12, pages 162–181, Berlin, Heidelberg, 2012. Springer.

[21] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity.* Springer, 2010.

[22] Rodney G. Downey, Carl Jockusch, and Michael Stob. Array nonrecursive sets and multiple permitting arguments. In *Recursion Theory Week: Proceedings of a Conference held in Oberwolfach, FRG, March 19-25, 1989*, pages 141–173. Springer, 1990.

[23] Rodney G. Downey, Denis R. Hirschfeldt, and Geoff LaForte. Randomness and reducibility. *Journal of Computer and System Sciences*, 68(1):96–114, 2004.

[24] Yun Fan and Liang Yu. Maximal pairs of c.e. reals in the computably Lipschitz degrees. *Annals of Pure and Applied Logic*, 162:357–366, 2011.

[25] Nan Fang and Wolfgang Merkle. Extending cl-reducibility on left-c.e. reals. In preparation.

[26] Péter Gács. Every sequence is reducible to a random one. *Information and Control*, 70(2):186–192, 1986.

[27] Ian Herbert. Lowness for integer-valued randomness. *Computability*, 5(2): 103–109, 2016.

[28] Antonín Kučera. Measure, Π^0_1-classes and complete extensions of PA. In Heinz-Dieter Ebbinghaus, Gert H. Müller, and Gerald E. Sacks, editors, *Recursion Theory Week: Proceedings of a Conference held in Oberwolfach, West Germany, April 15-21, 1984*, pages 245–259. Springer, 1985.

[29] Leonid A. Levin. *Some Theorems on the Algorithmic Approach to Probability Theory and Information Theory*. Dissertation in mathematics, Moscow University, 1971.

[30] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, third edition, 2008.

[31] Elliott H. Lieb, Daniel Osherson, and Scott Weinstein. Elementary proof of a theorem of Jean Ville. Unpublished manuscript: arXiv:cs/0607054v1, 2006.

[32] Jack H. Lutz. Gales and the constructive dimension of individual sequences. In *Automata, languages and programming (Geneva, 2000)*, volume 1853 of *Lecture Notes in Computer Science*, pages 902–913. Springer, Berlin, 2000.

[33] Jack H. Lutz. The dimensions of individual strings and sequences. *Information and Computation*, 187, 2003.

[34] Roger Mansuy. Histoire de martingales [the origins of the word 'martingale']. *Mathématiques & Sciences Humaines [Mathematical Social Sciences]*, 169(1):105–113, 2005. Translated from the French by Ronald Sverdlove, in Electronic Journal for History of Probability and Statistics. Vol.5, no.1, June 2009.

[35] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[36] Elvira Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Information Processing Letters*, 84(1):1–3, 2002.

[37] Wolfgang Merkle and Nenad Mihailović. On the construction of effectively random sets. *The Journal of Symbolic Logic*, 69(3):862–878, 09 2004.

[38] André Nies. *Computability and Randomness.* Oxford University Press, 2009.

[39] Piergiorgio Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers.* Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1992.

[40] Ron Peretz. Effective martingales with restricted wagers. *Information and Computation*, 245:152–164, 2015.

[41] Ron Peretz and Gilad Bavly. How to gamble against all odds. *Games and Economic behavior*, 94:157–168, 2015.

[42] Claus-Peter Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5(3):246–258, 1971.

[43] Claus-Peter Schnorr. *Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie.* Springer, Berlin, 1971. Lecture Notes in Mathematics, Vol. 218.

[44] Claus-Peter Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences*, 7(4):376–388, 1973.

[45] J. Laurie Snell. Gambling, probability and martingales. *The Mathematical Intelligencer*, 4(3):118–124, 1982.

[46] Robert I. Soare. Computability theory and differential geometry. *Bulletin of Symbolic Logic*, 10(4):457–486, 2004.

[47] Robert M. Solovay. Handwritten manuscript related to Chaitin's work. IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215 pages, 1975.

[48] Kohtaro Tadaki. A generalization of Chaitin's halting probability Ω and halting self-similar sets. *Hokkaido Mathematical Journal*, 31(1):219–253, 2002.

[49] Jason Teutsch. A savings paradox for integer-valued gambling strategies. *International Journal of Game Theory*, 43(1):145–151, 2014.

[50] Michiel van Lambalgen. *Random Sequences.* PhD dissertation, University of Amsterdam, The Netherlands, 1987.

[51] Jean-André Ville. *Étude Critique de la Notion de Collectif.* Gauthier-Villars, Paris, 1939. Monographies des Probabilités. Calcul des Probabilités et ses Applications.

[52] Liang Yu and Decheng Ding. There is no sw-complete ce real. *The Journal of Symbolic Logic*, 69(4):1163–1170, 2004.