Why Computational Complexity Requires Stricter Martingales[*]

John M. Hitchcock[†] Jack H. Lutz[‡]

Abstract

The word "martingale" has related, but different, meanings in probability theory and theoretical computer science. In computational complexity and algorithmic information theory, a martingale is typically a function d on strings such that $\mathrm{E}(d(wb)|w) = d(w)$ for all strings w, where the conditional expectation is computed over all possible values of the next symbol b. In modern probability theory a martingale is typically a sequence $\xi_0, \xi_1, \xi_2, \ldots$ of random variables such that $\mathrm{E}(\xi_{n+1}|\xi_0, \ldots, \xi_n) = \xi_n$ for all n.

This paper elucidates the relationship between these notions and proves that the latter notion is too weak for many purposes in computational complexity, because under this definition every computable martingale can be simulated by a polynomial-time computable martingale.

1 Introduction

Since martingales were introduced by Ville [22] in 1939 (having been implicit in earlier works of Lévy [9, 10]), they have followed two largely disjoint paths of scientific development and application. Along the larger and, to date, more significant path, Doob developed them into a powerful tool of probability theory that, especially following his influential 1953 book [6], has become central to many areas of research, including probability, stochastic processes, functional analysis, fractal geometry, statistical mechanics, and mathematical finance. Along the smaller and more recent path, effective martingales (martingales satisfying various computability conditions) have been used in theoretical computer science, first in the 1970's by Schnorr [18, 19, 20, 21] in his investigations of Martin-Löf's definition of randomness [15] and variants thereof, and then in the 1990's by Lutz [12, 14] in the development of resource-bounded measure. Many researchers have extended these developments, and effective martingales are now an active research topic that makes frequent contributions to our understanding of computational complexity, randomness, and algorithmic information.

[*] This research was supported in part by National Science Foundation Grants 9988483 and 0344187.

[†] Department of Computer Science, University of Wyoming, Laramie, WY 82071. jhitchco@cs.uwyo.edu.

[‡] Department of Computer Science, Iowa State University, Ames, IA 50011. lutz@cs.iastate.edu.

A curious thing about these two paths of research is that they interpret the word "martingale" differently. In computational complexity and algorithmic information theory, a martingale is typically a real-valued function d on $\{0,1\}^*$ such that

$$\mathrm{E}[d(wb)|w] = d(w) \tag{1.1}$$

for all strings w, where the expectation is conditioned on the *bit history* w (the string seen thus far) and computed over the two possible values of the next bit b. When the underlying probability measure is uniform (0 and 1 equally likely, independent of prior history), equation (1.1) becomes the familiar identity

$$d(w) = \frac{d(w0) + d(w1)}{2}. \tag{1.2}$$

Intuitively, a martingale d is a strategy for betting on the successive bits of an infinite binary sequence, and $d(w)$ is the amount of capital that a gambler using d will have after w if the sequence starts with w. Thus $d(\lambda)$ is the initial capital, and equation (1.1) says that the payoffs are fair.

On the other hand, in probability theory, a martingale is typically a sequence $\xi_0, \xi_1, \xi_2, \ldots$ of random variables such that

$$\mathrm{E}[\xi_{n+1}|\xi_0, \ldots, \xi_n] = \xi_n \tag{1.3}$$

for all $n \in \mathbb{N}$. Such a sequence is also called a martingale sequence or a *martingale process*, and we exclusively use the latter term here in order to distinguish the two notions under discussion.

To understand the essential (i.e., essential and nonobvious) difference between martingales and martingale processes, we first need to dispose of three inessential differences. First a martingale is a function from $\{0,1\}^*$ to $\mathbb{R}$, while a martingale process is a sequence of random variables. To see that this is only a difference in notation, let $\mathbf{C}$ be the Cantor space, consisting of all infinite binary sequences. Then we can identify each martingale d with the sequence $\xi_0, \xi_1, \xi_2, \ldots$ of functions $\xi_n : \mathbf{C} \to \mathbb{R}$ defined by

$$\xi_n(S) = d(S[0..n-1]),$$

where $S[0..n-1]$ is the n-bit prefix of S. Then $\xi_0, \xi_1, \xi_2, \ldots$ is a sequence of random variables and equation (1.1) says that

$$\mathrm{E}[\xi_{n+1}|w] = \xi_n \tag{1.4}$$

for all $n \in \mathbb{N}$ and $w \in \{0,1\}^n$. (See sections 2 and 3 for a precise treatment of this and other ideas developed intuitively in this introduction.)

The other two inessential differences are that martingales, unlike martingale processes, are typically required to be nonnegative and to have $\mathbf{C}$ as their underlying sample space (i.e., as the domain of each of the random variables ξ_n). To date it has been convenient to include nonnegativity in the martingale definition because most applications have required martingales that are nonnegative (or, equivalently, bounded below). Similarly, it has been convenient to have $\mathbf{C}$ – or some similar sequence space – as the underlying sample space because martingales have been used to investigate the structures of such spaces. However, the first of these requirements has an obvious effect, not needing further analysis, while the second is inessential and unlikely to persist into the future. In this paper, in order to facilitate our comparison, we ignore the nonnegativity requirement on martingales, and for both martingales and martingale processes, we focus on the case where the underlying sample space is $\mathbf{C}$.

The essential difference between the martingale processes of probability theory and the martingales of theoretical computer science is thus the difference between equations (1.3) and (1.4). Translating our remarks following (1.1) into the notation of (1.4), ξ_n denotes the gambler's capital after n bets, and equation (1.4) says that for each bit history $w \in \{0,1\}^n$, the expected value of the gambler's capital ξ_{n+1} after the next bet, *conditioned on the bit history* w, is the gambler's capital ξ_n before the next bet. In contrast, equation (1.3) says that for each *capital history* $c_0, \ldots, c_n$, the expected value of the gambler's capital ξ_{n+1} after the next bet, *conditioned on the capital history* $\xi_0 = c_0, \ldots, \xi_n = c_n$, is the gambler's capital ξ_n before the next bet. As we shall see, it is clear that (1.3) holds if (1.4) holds, but if two or more bit histories correspond to the same capital history, then it is possible to satisfy (1.3) without satisfying (1.4). Thus the martingale requirement of theoretical computer science is *stricter* than the martingale process requirement of probability theory.

In this paper we prove that this strictness is essential for computational complexity in the sense that martingale processes cannot be used in place of martingales as a basis for resource-bounded measure or resource-bounded randomness.

Resource-bounded measure uses resource-bounded martingales to define measure in complexity classes [12, 13, 14]. For example, a set X of decision problems has *measure 0 in* the complexity class E = DTIME(2^{linear}), and we write $\mu(X|\mathrm{E}) = 0$, if there is a polynomial time computable nonnegative martingale that *succeeds*, i.e., wins an unbounded amount of money on, every element of $X \cap \mathrm{E}$. An essential condition for this definition to be nontrivial is that E does not have measure 0 in itself, i.e., that there is no polynomial-time nonnegative martingale that succeeds on every element of E. This is indeed true by the Measure Conservation Theorem [12].

In contrast, we show here that there *is* a polynomial-time nonnegative martingale *process* that succeeds on every element of E. In fact, our main theorem says that for *any* computable nonnegative martingale process d, there is a polynomial-time nonnegative martingale process d' that succeeds on every sequence that d succeeds on. That is, computable nonnegative martingale processes cannot use time beyond polynomial to succeed on additional sequences. It follows that for every subclass $\mathcal{C}$ of every computably presentable class of decision problems – and hence for every reasonable uniform complexity class $\mathcal{C}$ – there is a polynomial-time nonnegative martingale process that succeeds on every element of $\mathcal{C}$. Thus martingale processes cannot be used as a basis for resource-bounded measure.

Martingale processes are similarly inadequate for resource-bounded randomness. For example, a sequence $S \in \mathbf{C}$ is p-*random* if there is no polynomial-time nonnegative martingale that succeeds on it [20, 12]. An essential feature of resource-bounded randomness is the existence [20], in fact abundance [12, 2], of decidable sequences that are random with respect to a given resource bound. For example, although no element of E can be p-random, almost every element of the complexity class EXP = DTIME($2^{\text{polynomial}}$) is p-random [12, 2]. However, the preceding paragraph implies that for every decidable sequence S there is a polynomial-time nonnegative martingale process that succeeds on S, so *no* decidable sequence could be p-random if we used martingale processes in place of martingales in defining p-randomness. Moreover, we also show that there exist computably random sequences (sequences on which no computable nonnegative martingale succeeds) on which polynomial-time nonnegative martingale processes can succeed.

Historically, the 1939 martingale definition of Ville [22] was the strict definition (1.4) now used in theoretical computer science. It was Doob [5] who in 1940 relaxed Ville's definition to the form (1.3) that is now so common in probability theory [7, 17, 1]. Of course the difference in usage between these two fields is not at all a dichotomy. The relaxed definition (1.3) is used in randomized algorithms [16] and other areas of theoretical computer science where the complexities of the martingales are not an issue, and probability theory also uses the more abstract notion of an $\vec{\mathcal{F}}$-martingale process (also formulated by Doob [5] and described in section 3 below), of which martingales and martingale processes are the two extreme cases.

Our results show that resource-bounded measure and randomness do in fact require martingales that are stricter than the martingale processes used so commonly in probability theory. However, these results do not disparage the latter notion. Quite to the contrary, it is to be anticipated that theoretical computer science will avail itself of and effectivize increasingly sophisticated aspects of martingales and measure-theoretic probability in the coming years. Our results and the arguments by which we prove them are to be regarded as steps toward expanding the interface between these two fields.

2 Preliminaries

A *decision problem* (a.k.a. *language*) is a set $A \subseteq \{0,1\}^*$. We identify each language with its characteristic sequence $[\![s_0 \in A]\!][\![s_1 \in A]\!][\![s_2 \in A]\!]\cdots$, where $s_0, s_1, s_2, \ldots$ is the standard enumeration of $\{0,1\}^*$ and $[\![\phi]\!] = \texttt{if } \phi \texttt{ then } 1 \texttt{ else } 0$. We write $A[i..j]$ for the string consisting of the i-th through j-th bits of (the characteristic sequence of) A.

A class $\mathcal{C}$ of languages is *computably presentable* (a.k.a. *recursively presentable* [3]) if there is an effective enumeration $M_0, M_1, \ldots$ of deterministic Turing machines, each of which halts on all inputs, such that $\mathcal{C} = \{L(M_i) | i \in \mathbb{N}\}$, where $L(M_i)$ is the language decided by M_i.

A *prefix set* is a language A such that no element of A is a prefix of any other element of A. If A is a language and $n \in \mathbb{N}$, then we write $A_{=n} = A \cap \{0,1\}^n$ and $A_{\leq n} = A \cap \{0,1\}^{\leq n}$.

The Cantor space $\mathbf{C}$ is the set of all infinite binary sequences. If $w \in \{0,1\}^*$ and $x \in \{0,1\}^* \cup \mathbf{C}$, then $w \sqsubseteq x$ means that w is a prefix of x. The *cylinder* generated by a string $w \in \{0,1\}^*$ is $\mathbf{C}_w = \{A \in \mathbf{C} \mid w \sqsubseteq A\}$.

A *σ-algebra* on $\mathbf{C}$ is a nonempty collection $\mathcal{F}$ of subsets of $\mathbf{C}$ that is closed under complements and under countable unions. For any collection $\mathcal{A}$ of subsets of $\mathbf{C}$ there is a unique smallest σ-algebra $\sigma(\mathcal{A})$ on $\mathbf{C}$ that contains $\mathcal{A}$. The Borel σ-algebra on $\mathbf{C}$ is $\mathcal{B} = \sigma(\{\mathbf{C}_w | w \in \{0,1\}^*\})$. We use the uniform probability measure on $\mathbf{C}$, which is the function $\mu : \mathcal{B} \to [0,1]$ determined by the values $\mu(w) = \mu(\mathbf{C}_w) = 2^{-|w|}$ for all $w \in \{0,1\}^*$.

Let $\mathcal{F}$ be a σ-algebra on $\mathbf{C}$. We say that a function $f : \mathbf{C} \to \mathbb{R}$ is *$\mathcal{F}$-measurable* if for all $t \in \mathbb{R}$,

$$\{S \in \mathbf{C} | f(S) \leq t\} \in \mathcal{F}.$$

A *random variable* on $\mathbf{C}$ is a function $\xi : \mathbf{C} \to \mathbb{R}$ that is $\mathcal{B}$-measurable. We write $\mathrm{E}[\xi]$ for the *expectation* of a random variable ξ. The *indicator function* of a set $A \subseteq \mathbf{C}$ is the function

$$1_A : \mathbf{C} \to \{0, 1\}$$

$$1_A(S) = \begin{cases} 1 & \text{if } S \in A \\ 0 & \text{if } S \notin A. \end{cases}$$

If ξ is a random variable and $A \subseteq \mathbf{C}$ satisfies $\mu(A) > 0$, then the *conditional expectation of ξ given A* is

$$\mathrm{E}[\xi|A] = \frac{\mathrm{E}[\xi \cdot 1_A]}{\mu(A)}.$$

If $\xi_0, \ldots, \xi_{n+1}$ are random variables and $t_0, \ldots, t_n \in \mathbb{R}$, we write $E[\xi_{n+1}|\xi_0 = t_0, \ldots, \xi_n = t_n]$ for $E[\xi_{n+1}|\{S \in \mathbf{C}|\xi_0(S) = t_0, \ldots, \xi_n(S) = t_n\}]$. If ξ is a random variable, $\mathcal{A}$ is a countable partition of $\mathbf{C}$, and $\mathcal{F} = \sigma(\mathcal{A})$, then the *conditional expectation of ξ given $\mathcal{F}$* is the random variable

$$\mathrm{E}[\xi|\mathcal{F}] : \mathbf{C} \to \mathbb{R}$$

$$\mathrm{E}[\xi|\mathcal{F}](S) = \mathrm{E}[\xi|A] \text{ where } S \in A \in \mathcal{A}$$

which is defined for μ-almost all S.

We say that a real-valued function $f : \{0,1\}^* \to \mathbb{R}$ is *computable* if there is a computable function $\hat{f} : \mathbb{N} \times \{0,1\}^* \to \mathbb{Q}$ such that for all $n \in \mathbb{N}$ and $w \in \{0,1\}^*$, $|f(w) - \hat{f}(n, w)| \leq 2^{-n}$. We say that $\hat{f}$ is a *computation* of f. We often write $\hat{f}_n(w)$ for $\hat{f}(n, w)$. If $\hat{f}$ is computable in polynomial-time (where n is input in unary), then f is *polynomial-time computable.* If $f : \{0,1\}^* \to \mathbb{Q}$ is itself a (polynomial-time) computable function, then we say that f is *(polynomial-time) exactly computable.*

We say that $f : \{0,1\}^* \to \mathbb{R}$ is *constructive* (a.k.a. *lower semicomputable*) if there is a computable function $h : \mathbb{N} \times \{0,1\}^* \to \mathbb{Q}$ such that for any $w \in \{0,1\}^*$, $h(n, w) \leq h(n+1, w) < f(w)$ for all $n \in \mathbb{N}$ and $\lim_{n\to\infty} h(n, w) = f(w)$.

3 Varieties of Martingales

In this section we introduce the different notions of martingales used in theoretical computer science and probability theory. As noted in the introduction, we use the terms "martingale" for the former and "martingale process" for the latter. We begin with the martingale definition commonly used in the theory of computing.

Definition. A function $d : \{0,1\}^* \to \mathbb{R}$ is a *martingale* if

$$d(w) = \frac{d(w0) + d(w1)}{2} \tag{3.1}$$

for all $w \in \{0,1\}^*$.

Intuitively, a martingale d represents a strategy in a betting game. The gambler begins with $d(\lambda)$ of capital and is betting on an unknown sequence $S \in \mathbf{C}$. The gambler places a wager on the first bit of S being 0 or 1. If the first bit of S is 0, the gambler then holds $d(0)$ capital; otherwise, the first bit is 1 and the gambler holds $d(1)$ capital. The gambler then bets on the second bit of S possibly using his knowledge of the first bit of S. In general, after n rounds of this game, the gambler knows that the first n bits of S are $w = S[0..n-1]$. Using this knowledge he wagers on the $(n+1)$-st bit of S. Equation (3.1) says that this is a fair gambling game. That is, the payoffs are fair: if S is chosen uniformly at random, the gambler can expect to have the same amount of capital after each stage of the game.

We will use random variables and conditional expectations to make this idea of fair gambling more precise. Let $d : \{0,1\}^* \to \mathbb{R}$ be an arbitrary function. For each $n \in \mathbb{N}$, we define the function

$$\xi_{d,n} : \mathbf{C} \to \mathbb{R}$$
$$\xi_{d,n}(S) = d(S[0..n-1]).$$

Observe that each $\xi_{d,n}$ is a discrete random variable on $(\mathbf{C}, \mathcal{B}, \mu)$. We associate the sequence of random variables $\vec{\xi_d} = (\xi_{d,0}, \xi_{d,1}, \ldots)$ with d. We can now interpret the martingale condition (3.1) as a conditional expectation.

Observation 3.1. *A function $d : \{0,1\}^* \to \mathbb{R}$ is a martingale if and only if*

$$\mathrm{E}[\xi_{d,|w|+1} | \mathbf{C}_w] = \xi_{d,|w|} \tag{3.2}$$

for all $w \in \{0,1\}^*$.

In probability theory, martingales are typically defined in the following more general form.

Definition. Let $\vec{\xi} = (\xi_0, \xi_1, \ldots)$ be a sequence of random variables. We say that $\vec{\xi}$ is a *martingale process* if for all $n \in \mathbb{N}$, $\mathrm{E}[\xi_n] < \infty$ and

$$\mathrm{E}[\xi_{n+1} | \xi_0 = c_0, \ldots, \xi_n = c_n] = c_n \tag{3.3}$$

for all values of $c_0, \ldots, c_n \in \mathbb{R}$. (As we shall see below, condition (3.3) can also be stated more concisely using a conditional expectation given a σ-algebra.)

We can also view a martingale process $\vec{\xi}$ as a gambling game. Again the gambler is wagering on an unknown sequence $S \in \mathbf{C}$. The initial capital is $\xi_0(S)$. After the n^{th} stage of the game, the gambler has capital $\xi_n(S)$. The condition (3.3) says that the payoffs are fair in this game. This notion of fairness is more relaxed than the martingale condition (3.2). In order to make a precise comparison we extend the definition of martingale processes.

Definition. A function $d : \{0,1\}^* \to \mathbb{R}$ is a *martingale process* if $\vec{\xi_d}$ is a martingale process.

The martingale process condition for a function d is

$$\mathrm{E}[\xi_{d,n+1} | \xi_{d,0} = c_0, \ldots, \xi_{d,n} = c_n] = c_n. \tag{3.4}$$

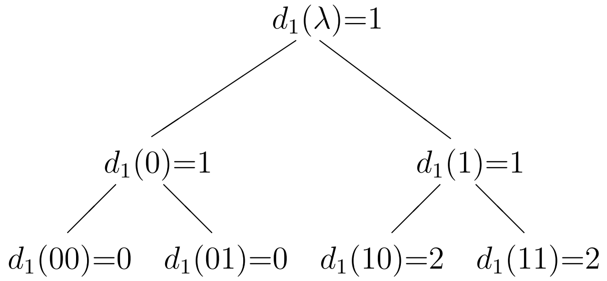

Figure 3.1: The martingale process d_1 of Example 3.3.

This fairness condition involves the *capital history* of the gambling game rather than revealed *bit history* of the sequence S. In (3.2), the conditioning is done on the bit history w. The conditioning in (3.4) is done on the capital history. Intuitively, the martingale condition is more "local" than the martingale process condition.

We now give a more concrete characterization of which functions $d : \{0,1\}^* \to \mathbb{R}$ are martingale processes. Define an equivalence relation $\approx_d$ on $\{0,1\}^*$ by

$$x \approx_d y \iff |x| = |y| \text{ and } (\forall 1 \leq i \leq |x|)\ d(x[0..i-1] = d(y[0..i-1]).$$

For each $w \in \{0,1\}^*$ we define the equivalence class $[w]_d = \{v \in \{0,1\}^* | w \approx_d v\}$.

Observation 3.2. *A function $d : \{0,1\}^* \to \mathbb{R}$ is a martingale process if and only if*

$$2\Big|[w]_d\Big|d(w) = \sum_{v \in [w]_d} d(v0) + d(v1) \tag{3.5}$$

for all $w \in \{0,1\}^*$.

Any martingale d is also a martingale process; the following example shows that the converse is not true.

Example 3.3. Define for all $u \in \{0,1\}^*$

$$\begin{aligned} &d_1(\lambda) = d_1(0) = d_1(1) = 1, \\ &d_1(0u) = 0, \\ &d_1(1u) = 2. \end{aligned}$$

Then d_1 is not a martingale, but d_1 is a martingale process. Because the strings 0 and 1 have the same capital histories, $[0]_{d_1} = \{0,1\}$, so the averaging condition (3.5) allows the capital to "shift" in a manner not allowed by a martingale.

We now discuss a more general formulation of martingales that is used in probability theory. See also [5, 6, 16, 4] for discussions of this notion. The following definition will yield the martingales and martingale processes defined above as special cases.

Definition. 1. A *filtration on* **C** is a sequence of σ-algebras $\vec{\mathcal{F}} = (\mathcal{F}_0, \mathcal{F}_1, \ldots)$ on **C** such that $\mathcal{F}_n \subseteq \mathcal{F}_{n+1}$ for all $n \in \mathbb{N}$.

2. Let $\vec{\xi}$ be a sequence of random variables and let $\vec{\mathcal{F}}$ be a filtration on $\mathbf{C}$. Then $\vec{\xi}$ is an *$\vec{\mathcal{F}}$-martingale process* if the following conditions hold.

   (i) For all $n \in \mathbb{N}$, ξ_n is $\mathcal{F}_n$-measurable and $\mathrm{E}[\xi_n] < \infty$.

   (ii) For all $n \in \mathbb{N}$,
   $$\mathrm{E}[\xi_{n+1} | \mathcal{F}_n] = \xi_n. \tag{3.6}$$

   We also say that $\vec{\xi}$ is a *martingale relative to $\vec{\mathcal{F}}$*.

The conditional expectation (3.6) can be viewed as a more generalized notion of fairness in the gambling game. For example, if $\vec{\xi}$ is an $\vec{\mathcal{F}}$-martingale process for some filtration $\vec{\mathcal{F}}$, then $\vec{\xi}$ is also a martingale process. Before we make any further comparisons we extend the filtration definition to functions.

Definition. Let $\vec{\mathcal{F}}$ be a filtration. A function $d : \{0,1\}^* \to \mathbb{R}$ is an *$\vec{\mathcal{F}}$-martingale process* if $\vec{\xi_d}$ is an $\vec{\mathcal{F}}$-martingale process.

For each $n \in \mathbb{N}$, let
$$\mathcal{M}_n = \sigma(\{\mathbf{C}_w | w \in \{0,1\}^n\}).$$
We let $\vec{\mathcal{M}} = (\mathcal{M}_0, \mathcal{M}_1, \ldots)$.

Observation 3.4. *A function $d : \{0,1\}^* \to \mathbb{R}$ is a martingale if and only if d is an $\vec{\mathcal{M}}$-martingale process.*

Let $d : \{0,1\}^* \to \mathbb{R}$ be arbitrary. For each $n \in \mathbb{N}$, define
$$B_{d,n} = \{[w]_d | w \in \{0,1\}^n\},$$
$$\mathcal{C}_{d,n} = \left\{ \bigcup_{w \in A} \mathbf{C}_w \,\middle|\, A \in B_{d,n} \right\}, \text{ and}$$
$$\mathcal{F}_{d,n} = \sigma(\mathcal{C}_{d,n}).$$
We let $\vec{\mathcal{F}_d} = (\mathcal{F}_{d,0}, \mathcal{F}_{d,1}, \ldots)$.

Observation 3.5. *A function $d : \{0,1\}^* \to \mathbb{R}$ is a martingale process if and only if d is an $\vec{\mathcal{F}_d}$-martingale process.*

If d is a martingale relative to some filtration $\vec{\mathcal{F}}$, then d is also an $\vec{\mathcal{F}_d}$-martingale process. That is, the martingale process requirement uses the coarsest filtration possible. On the other hand, the martingale requirement uses the essentially finest filtration $\vec{\mathcal{M}}$. (If $\vec{\mathcal{F}}$ is a finer filtration than $\vec{\mathcal{M}}$, then d is an $\vec{\mathcal{F}}$-martingale process if and only if d is an $\vec{\mathcal{M}}$-martingale process.)

A very useful property of martingales in theoretical computer science is that the sum of two martingales is a martingale. For any filtration $\vec{\mathcal{F}}$, the analogous fact also holds for $\vec{\mathcal{F}}$-martingale processes. In contrast, it is well known [4] that the sum of two martingale processes need not be a martingale process. We include an example for completeness.

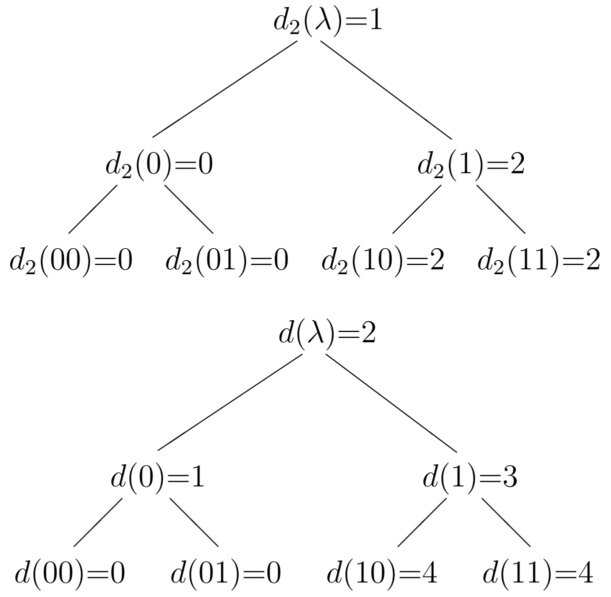

Figure 3.2: The martingale d_2 and the function d of Example 3.6.

Example 3.6. Define for all $u \in \{0,1\}^*$ and $v \in \{0,1\}^+$

$$d_2(\lambda) = 1,\ d_2(0u) = 0,\ d_2(1u) = 2,$$

and

$$d(\lambda) = 2,\ d(0) = 1,\ d(1) = 3,\ d(0v) = 0,\ d(1v) = 4.$$

Then d_2 is martingale, so it is a martingale process. Let d_1 be the martingale process from Example 3.3. Then $d = d_1 + d_2$, but d is not a martingale process.

4 Martingale Processes and Complexity

In this section we present our results, all of which concern the complexities and success sets of martingale processes.

Definition. Let $d : \{0,1\}^* \to \mathbb{R}$.

1. We say that d *succeeds on* a sequence $S \in \mathbf{C}$ if $\limsup_{n\to\infty} d(S[0..n-1]) = \infty$.
2. The *success set* of d is $S^\infty[d] = \{S \in \mathbf{C} | d \text{ succeeds on } S\}$.

The following technical lemma is crucial for our main theorem.

Lemma 4.1. (Exact Computation Lemma) *For every computable martingale process d and every $m \in \mathbb{N}$, there is an exactly computable martingale process d' such that for all $w \in \{0,1\}^*$, $|d'(w) - d(w)| < 2^{-m}$.*

Proof. Let d and m be as given, and let $\hat{d}$ be a computation of d. We define $d' : \{0,1\}^* \to \mathbb{Q}$ by recursion on the lengths of strings.

At length 0 we set $d'(\lambda) = \hat{d}_{m+1}(\lambda)$.

Assume that $d'(w)$ has been defined for all $w \in \{0,1\}^{\leq n}$. Then we define a reflexive, symmetric relation $\sim$ on $\{0,1\}^{n+1}$ by

$$x \sim y \iff \left[x' \approx_{d'} y' \text{ and } |\hat{d}_{r+1}(x) - \hat{d}_{r+1}(y)| \leq 2^{-r} \right],$$

where x', y' are the n-bit prefixes of x, y, respectively, and $r = m + 2n + 5$. We then let $\approx$ be the transitive closure of $\sim$, noting that $\approx$ is an equivalence relation on $\{0,1\}^{n+1}$. For each $w \in \{0,1\}^{n+1}$, let

$$d''(w) = \operatorname*{avg}_{v \approx w} \hat{d}_{r+1}(v), \tag{4.1}$$

where "avg" denotes the arithmetic mean. Finally, for each $u \in \{0,1\}^n$ and $b \in \{0,1\}$, let

$$\Delta u = d'(u) - \operatorname*{avg}_{\substack{v \approx_{d'} u \\ b \in \{0,1\}}} d''(vb) \tag{4.2}$$

and

$$d'(ub) = d''(ub) + \Delta u. \tag{4.3}$$

This completes the definition of d'.

It is clear that d' is exactly computable. Also, for all $u \in \{0,1\}^*$, (4.3) and (4.2) ensure that

$$\operatorname*{avg}_{\substack{v \approx_{d'} u \\ b \in \{0,1\}}} d'(vb) = d'(u),$$

whence d' is a martingale process.

We now note four things about the construction of d'. First, for all $x, y \in \{0,1\}^*$, it is clear that

$$x \approx_d y \Rightarrow x \approx y. \tag{4.4}$$

Second, the triangle inequality and the fact that there are only 2^{n+1} strings in $\{0,1\}^{n+1}$ tell us that for all $x, y \in \{0,1\}^{n+1}$,

$$x \approx y \Rightarrow |\hat{d}_{r+1}(x) - \hat{d}_{r+1}(y)| \leq (2^{n+1} - 1)2^{-r}. \tag{4.5}$$

By (4.1) and (4.5), then, we have

$$|d''(w) - \hat{d}_{r+1}(w)| \leq (2^{n+1} - 1)2^{-r},$$

whence by the triangle inequality,

$$|d''(w) - d(w)| \leq 2^{n+1-r} \tag{4.6}$$

for all $w \in \{0,1\}^{n+1}$. Third, for all $x, y \in \{0,1\}^{n+1}$,

$$x \approx y \Rightarrow d''(x) = d''(y). \tag{4.7}$$

Fourth, for all $u, v \in \{0,1\}^n$,

$$u \approx_{d'} v \Rightarrow \Delta u = \Delta v, \tag{4.8}$$

from which (4.3) tells us that for all $x, y \in \{0,1\}^{n+1}$,

$$x \approx_{d'} y \iff d''(x) = d''(y). \tag{4.9}$$

To complete the proof it suffices to show that

$$|d'(u) - d(u)| \leq 2^{-m}(1 - 2^{-(|u|+1)}) \tag{4.10}$$

holds for all $u \in \{0,1\}^*$. We prove this by induction on u. Since

$$|d'(\lambda) - d(\lambda)| \leq 2^{-(m+1)} = 2^{-m}(1 - 2^{-(0+1)}),$$

it is clear that (4.10) holds for λ. Assume that (4.10) holds for u, let $n = |u|$, and let $b \in \{0,1\}$. Then by (4.3) and (4.6) we have

$$|d'(ub) - d(ub)| \leq |d''(ub) - d(ub)| + |\Delta u| \leq 2^{n+1-r} + |\Delta u|. \tag{4.11}$$

Also, by (4.2) and the induction hypothesis, we have

$$\begin{aligned} |\Delta u| \leq & \quad |d'(u) - d(u)| + \left| d(u) - \operatorname*{avg}_{\substack{v \approx_{d'} u \\ b \in \{0,1\}}} d''(vb) \right| \\ \leq & \quad 2^{-m}(1 - 2^{-(n+1)}) + \left| d(u) - \operatorname*{avg}_{\substack{v \approx_{d'} u \\ b \in \{0,1\}}} d''(vb) \right| \end{aligned} \tag{4.12}$$

We now have two cases.

Case I. $[u]_{d'} = [u]_d$. Then, since d is a martingale process, (4.6) tells us that

$$\begin{aligned} \left| d(u) - \operatorname*{avg}_{\substack{v \approx_{d'} u \\ b \in \{0,1\}}} d''(vb) \right| &= \left| \operatorname*{avg}_{\substack{v \approx_{d} u \\ b \in \{0,1\}}} d(vb) - \operatorname*{avg}_{\substack{v \approx_{d} u \\ b \in \{0,1\}}} d''(vb) \right| \\ &\leq \operatorname*{avg}_{\substack{v \approx_{d} u \\ b \in \{0,1\}}} |d(vb) - d''(vb)| \\ &\leq 2^{n+1-r}. \end{aligned}$$

Case II. $[u]_{d'} \neq [u]_d$. Then $n > 0$ and by (4.4), (4.7), and (4.9) there exist $u_1, \ldots, u_k \in \{0,1\}^n$ such that $\{[u_1]_d, \ldots, [u_k]_d\}$ is a partition of $[u]_{d'}$. For each $1 \leq i \leq k$, (4.9) and (4.6) tell us that

$$\begin{aligned} |d(u) - d(u_i)| &\leq |d(u) - d''(u)| + |d''(u) - d(u_i)| \\ &= |d(u) - d''(u)| + |d''(u_i) - d(u_i)| \\ &\leq 2 \cdot 2^{n-r} \\ &= 2^{n+1-r}, \end{aligned}$$

whence

$$\left| d(u) - \operatorname*{avg}_{1\le i\le k} d(u_i) \right| \le 2^{n+1-r}. \tag{4.13}$$

Also, since d is a martingale process, (4.6) tells us that

$$\begin{aligned}
\left| \operatorname*{avg}_{1\le i\le k} d(u_i) - \operatorname*{avg}_{\substack{v\approx_{d'}u \\ b\in\{0,1\}}} d''(vb) \right| &= \left| \operatorname*{avg}_{1\le i\le k} d(u_i) - \operatorname*{avg}_{1\le i\le k} \operatorname*{avg}_{\substack{v\approx_{d}u_i \\ b\in\{0,1\}}} d''(vb) \right| \\
&\le \operatorname*{avg}_{1\le i\le k} \left| d(u_i) - \operatorname*{avg}_{\substack{v\approx_{d}u_i \\ b\in\{0,1\}}} d''(vb) \right| \\
&= \operatorname*{avg}_{1\le i\le k} \left| \operatorname*{avg}_{\substack{v\approx_{d}u_i \\ b\in\{0,1\}}} d(vb) - \operatorname*{avg}_{\substack{v\approx_{d}u_i \\ b\in\{0,1\}}} d''(vb) \right| \\
&\le \operatorname*{avg}_{1\le i\le k} \operatorname*{avg}_{\substack{v\approx_{d}u_i \\ b\in\{0,1\}}} |d(vb) - d''(vb)| \\
&\le 2^{n+1-r}.
\end{aligned}$$

Combined with (4.13), this tells us that

$$\left| d(u) - \operatorname*{avg}_{\substack{v\approx_{d'}u \\ b\in\{0,1\}}} d''(vb) \right| \le 2^{n+2-r}.$$

In either Case I or Case II, (4.12) tells us that

$$|\Delta u| \le 2^{-m}(1 - 2^{-(n+1)}) + 2^{n+2-r},$$

whence (4.11) and our choice of r tell us that

$$\begin{aligned}
|d'(ub) - d(ub)| &< 2^{-m}(1 - 2^{-(n+1)}) + 2^{n+3-r} \\
&= 2^{-m}(1 - 2^{-(n+1)}) + 2^{-(m+n+2)} \\
&= 2^{-m}(1 - 2^{-(n+2)}),
\end{aligned}$$

i.e., (4.10) holds for ub. □

Our next lemma can be regarded as a "speedup" theorem for exactly computable martingale processes, but its proof uses a very slow simulation technique analogous to slow diagonalization.

Lemma 4.2. *For every exactly computable nonnegative martingale process d there is a polynomial-time exactly computable nonnegative martingale process d' such that $S^\infty[d] = S^\infty[d']$.*

Proof. Let d be an exactly computable martingale process. Consider an algorithm that on input w of length n computes $d(v)$ for all strings v in standard ordering until it has used n^2 computation steps. Let $m(n)$ be the largest integer such that $d(v)$ is computed for all strings of length $m(n)$ by this algorithm and choose N such that $m(N) > 0$. We define

$$d' : \{0,1\}^* \to \mathbb{R}$$

$$d'(w) = \begin{cases} d(\lambda) & \text{if } |w| < N \\ d(w[0..m(|w|)-1]) & \text{if } |w| \geq N. \end{cases}$$

Then d' is a polynomial-time exactly computable martingale process and $S^\infty[d] = S^\infty[d']$. □

We now have the main theorem of this paper, which says that polynomial-time computable martingale processes are equivalent to arbitrary computable martingale processes.

Theorem 4.3. *For every computable nonnegative martingale process d there is a polynomial-time exactly computable nonnegative martingale process d' such that $S^\infty[d] = S^\infty[d']$.*

Proof. This follows immediately from Lemmas 4.1 and 4.2. □

Theorem 4.3 has the following consequence for resource-bounded measure.

Corollary 4.4. *For every computably presentable class $\mathcal{C}$, there is a polynomial-time exactly computable nonnegative martingale process d such that $\mathcal{C} \subseteq S^\infty[d]$.*

Proof. Lutz [12] has shown that for every computably presentable class $\mathcal{C}$ (called "rec-countable" in the terminology of [12]) there is a computable nonnegative martingale d such that $\mathcal{C} \subseteq S^\infty[d]$. Since d is a computable martingale process, the conclusion of the corollary follows by Theorem 4.3. □

Since complexity classes such as $\mathrm{E}, \mathrm{EXP}, \mathrm{ESPACE}$, etc. are all computably presentable, Corollary 4.4 implies that martingale processes cannot be used in place of martingales as a basis for resource-bounded measure.

We now prove a generalized Kraft inequality that enables us to establish an upper bound on the power of computable martingale processes. For any function $d : \{0,1\}^* \to \mathbb{R}$ and $A \subseteq \{0,1\}^*$, we say that A is *closed under* $\approx_d$ if for all $w \in \{0,1\}^*$,

$$w \in A \Rightarrow [w]_d \subseteq A.$$

Lemma 4.5. *If d is a nonnegative martingale process and $A \subseteq \{0,1\}^*$ is a prefix set that is closed under $\approx_d$, then*

$$\sum_{w \in A} d(w) 2^{-|w|} \leq d(\lambda).$$

Proof. We first use induction on n to prove that for all $n \in \mathbb{N}$, the lemma holds for all prefix sets $A \subseteq \{0,1\}^{\leq n}$ that are closed under $\approx_d$. For $n = 0$, this is trivial. Assume that it holds for n, and let $A \subseteq \{0,1\}^{\leq n+1}$ be a prefix set that is closed under $\approx_d$. Let

$$A' = \{w \in \{0,1\}^n \mid w0 \in A \text{ or } w1 \in A\} ,$$

$$A'' = \{v \in \{0,1\}^n \mid (\exists w \in A')v \approx_d w\},$$
$$B = \{w \in A'' \mid (\forall v \in [w]_d)\ w \leq v\},$$

and let

$$C = A_{\leq n} \cup A''.$$

Note that C is a prefix set and C is closed under $\approx_d$. Also, $A_{\leq n} \cap A' = \emptyset$ because A is a prefix set, so $A_{\leq n} \cap A'' = \emptyset$ because A is closed under $\approx_d$. Also,

$$\begin{aligned}
\sum_{w \in A_{=n+1}} 2^{-|w|} d(w) &= 2^{-(n+1)} \sum_{w \in A_{=n+1}} d(w) \\
&\leq 2^{-(n+1)} \sum_{w \in A'} [d(w0) + d(w1)] \\
&\leq 2^{-(n+1)} \sum_{w \in A''} [d(w0) + d(w1)] \\
&= 2^{-(n+1)} \sum_{w \in B} \sum_{v \in [w]_d} [d(v0) + d(v1)] \\
&= 2^{-(n+1)} \sum_{w \in B} 2 \Big| [w]_d \Big| d(w) \\
&= 2^{-n} \sum_{w \in A''} d(w).
\end{aligned}$$

Since $C \subseteq \{0,1\}^{\leq n}$, it follows by the induction hypothesis that

$$\begin{aligned}
\sum_{w \in A} 2^{-|w|} d(w) &= \sum_{w \in A_{\leq n}} 2^{-|w|} d(w) + \sum_{w \in A_{=n+1}} 2^{-|w|} d(w) \\
&\leq \sum_{w \in A_{\leq n}} 2^{-|w|} d(w) + \sum_{w \in A''} 2^{-|w|} d(w) \\
&= \sum_{w \in C} 2^{-|w|} d(w) \\
&\leq d(\lambda).
\end{aligned}$$

This completes the proof that for all $n \in \mathbb{N}$, the lemma holds for all prefix sets $A \subseteq \{0,1\}^{\leq n}$ that are closed under $\approx_d$.

To complete the proof of the lemma, let A be an arbitrary prefix set that is closed under $\approx_d$. Then

$$\sum_{w \in A} 2^{-|w|} d(w) = \sup_{n \in \mathbb{N}} \sum_{w \in A_{\leq n}} 2^{-|w|} d(w) \leq d(\lambda).$$

□

Theorem 4.6. *For every computable nonnegative martingale process d there is a constructive nonnegative martingale d' such that $S^\infty[d] \subseteq S^\infty[d']$.*

Proof. By Lemma 4.1 we may assume that d is exactly computable. Without loss of generality we also assume that $d(\lambda) \leq 1$. For each $k \in N$, let

$$A_k = \left\{ w \in \{0,1\}^* \,\middle|\, \max_{0 \leq i < |w|} d(w[0..i-1]) < 2^k \leq d(w) \right\}.$$

Then each A_k is a prefix set that is closed under $\approx_d$, so Lemma 4.5 implies that

$$\sum_{w \in A_k} 2^{-|w|} 2^k \leq \sum_{w \in A_k} 2^{-|w|} d(w) \leq d(\lambda) \leq 1;$$

hence

$$\sum_{w \in A_k} 2^{-|w|} \leq 2^{-k}.$$

For each $k \in \mathbb{N}$ we define a function

$$d'_k : \{0,1\}^* \to [0, \infty)$$

$$d'_k(w) = \begin{cases} 1 & \text{if } (\exists v \in A_k) v \sqsubseteq w \\ \sum\limits_{\substack{v \in A_k \\ w \sqsubseteq v}} 2^{|w|-|v|} & \text{otherwise.} \end{cases}$$

Next we define

$$d' : \{0,1\}^* \to [0, \infty)$$

$$d' = \sum_{k=0}^{\infty} d'_k.$$

As each d'_k is a martingale and

$$d'(\lambda) = \sum_{k=0}^{\infty} \sum_{w \in A_k} 2^{-|w|} \leq \sum_{k=0}^{\infty} 2^{-k} = 2,$$

d' is a martingale. Also, d' is constructive because the set

$$A = \{\langle k, w \rangle \mid \; w \in A_k\}$$

is decidable.

For each $k \in \mathbb{N}$, let

$$\mathcal{A}_k = \bigcup_{w \in A_k} \mathbf{C}_w.$$

For all $S \in \mathcal{A}_k$ there is some n_k such that $d'_k(S[0..n-1]) = 1$ for all $n \geq n_k$. For each $m \in \mathbb{N}$ and $S \in \bigcap_{k=0}^{m} \mathcal{A}_k$, $d'(S[0..n-1]) \geq \sum_{i=0}^{k} d'_k(S[0..n-1]) \geq k+1$ for all $n \geq \max\{n_0, \ldots, n_k\}$, so

$$S^\infty[d] = \bigcap_{k=0}^{\infty} \mathcal{A}_k \subseteq S^\infty[d'].$$

□

Theorem 4.6 implies that no computable nonnegative martingale process can succeed on a sequence that is random in the sense of Martin-Löf [15]. In contrast, we now show that there exist computably random sequences (sequences on which no computable nonnegative martingale succeeds [20]) on which polynomial time nonnegative martingale processes can succeed. In the following theorem $K(x)$ denotes the Kolmogorov complexity of x as defined in [11].

Theorem 4.7. *For every $S \in \mathbf{C}$ satisfying $K(S[0..n-1]) < n - \log n$ almost everywhere, there is a polynomial-time nonnegative martingale process that succeeds on S.*

Proof. We begin by fixing some suitable constants. Choose $c_0, c_1 \in \mathbb{N}$ such that

$$K(v) \leq K(uv) + K(|u|) + c_0 \tag{4.14}$$

for all $u, v \in \{0,1\}^*$ and

$$K(n) \leq \log n + c_1 \tag{4.15}$$

for all $n \in \mathbb{N}$. Let $c = c_0 + c_1$ and choose $N_0 \in \mathbb{N}$ such that for all $n \geq N_0$,

$$\log^{(2)} n + \log^{(3)} n + c < \log n - 1. \tag{4.16}$$

Let $S \in \mathbf{C}$ be such that $K(S[0..n-1]) < n - \log n$ for all $n \geq N_1$ and set $N = \max\{N_0, N_1\}$.

We now describe an inductive procedure that will be used to build a martingale process that succeeds on S. Initially, we let

$$\begin{aligned} m_0 &= N, \\ M_0 &= m_0, \\ W_{0,0} &= \{0,1\}^N, \\ a_0 &= 0, \\ C_{0,0} &= 2^N. \end{aligned}$$

Now assume that $m_i, M_i, W_{i,k}, a_i$, and $C_{i,k}$ have been defined for all $0 \leq i \leq k$. We use a dovetailing procedure to compute some string $x_{k+1} \in W_{0,k} \cup \cdots \cup W_{k,k}$ satisfying $K(x_{k+1}) < |x_{k+1}| - \log |x_{k+1}|$. Let i be the index of the set $W_{i,k}$ containing x_{k+1}. Now we set

$$\begin{aligned} W_{j,k+1} &= W_{j,k}, \\ C_{j,k+1} &= C_{j,k} \end{aligned}$$

for all $j \in \{0, \ldots, k\} - \{i\}$,

$$\begin{aligned} W_{i,k+1} &= W_{i,k} - \{x_{k+1}\}, \\ C_{i,k+1} &= C_{i,k} - 2(a_i + 1), \end{aligned}$$

and

$$\begin{aligned} M_{k+1} &= M_k + 2^{2^{M_k}}, \\ m_{k+1} &= M_{k+1} - |x_{k+1}|, \\ W_{k+1,k+1} &= x_{k+1} \cdot \{0,1\}^{m_{k+1}}, \\ a_{k+1} &= a_i + 1, \\ C_{k+1,k+1} &= (2a_{k+1})2^{m_{k+1}}. \end{aligned}$$

We define a function $d : \{0,1\}^* \to \mathbb{R}$ as follows. For all $w \in \{0,1\}^{<M_1}$, we let $d(w) = 1$. For all $k \geq 1$ and $w \in \{0,1\}^{<M_{k+1}} - \{0,1\}^{<M_k}$, we compute the unique index $i \in \{0, \ldots, k\}$ such that w has a prefix in $W_{i,k}$, and we let

$$d(w) = \frac{C_{i,k}}{|W_{i,k}|}.$$

Then d is an exactly computable function.

We claim that d is a nonnegative function. By a counting argument we know that for all n,

$$\left|\left\{v \in \{0,1\}^n \middle| K(v) < n - 1\right\}\right| \leq 2^{n-1}. \tag{4.17}$$

In particular, there are fewer than $2^{m_0 - 1}$ strings $w \in \{0,1\}^{m_0}$ satisfying $K(w) < |w| - \log|w|$, so $C_{0,j} \geq 0$ for all $j \in \mathbb{N}$. Now let $k \geq 1$. Let $v \in \{0,1\}^{m_k}$ and suppose that

$$K(x_k v) < |x_k v| - \log|x_k v|.$$

Notice that $m_k \geq 2^{2^{|x_k|}}$, so $|x_k| \leq \log^{(2)}|v|$. Applying (4.14), (4.15), and (4.16), we have

$$\begin{array}{rcl} K(v) & \leq & K(x_k v) + K(|x_k|) + c_0 \\ & < & |x_k v| - \log|x_k v| + K(|x_k|) + c_0 \\ & \leq & |x_k v| - \log|x_k v| + \log|x_k| + c \\ & \leq & |v| + \log^{(2)}|v| - \log|v| + \log^{(3)}|v| + c \\ & < & |v| - 1 \end{array}$$

This together with (4.17) implies that there are no more than $2^{m_k - 1}$ strings $w \in W_{k,k}$ for which $K(w) < |w| - \log|w|$. Hence $C_{k,j} > 0$ for all $j \geq k$, so d is a nonnegative function.

We claim that d is a martingale process. For all $w \in \{0,1\}^{<M_1 - 1}$, it is clear that the martingale process condition holds. Suppose it holds for all $w \in \{0,1\}^{<M_k - 1}$. In the k^{th} stage of the above procedure, for some i an $x_k \in W_{i,k-1}$ is chosen. Let $w \in \{0,1\}^{M_k - 1}$ and suppose that w has a prefix in $W_{i,k-1}$. Then

$$\begin{array}{rcl} \displaystyle\sum_{v \in [w]_d} d(v0) + d(v1) & = & \displaystyle\sum_{\substack{u \in W_{i,k-1} \\ v \in \{0,1\}^{m_k}}} d(uv) \\ & = & \displaystyle\sum_{\substack{u \in W_{i,k-1} - \{x_k\} \\ v \in \{0,1\}^{m_k}}} d(uv) + \sum_{v \in \{0,1\}^{m_k}} d(x_k v) \\ & = & (|W_{i,k-1}| - 1)2^{m_k} \dfrac{C_{i,k}}{|W_{i,k}|} + 2^{m_k} \dfrac{C_{k,k}}{|W_{k,k}|} \\ & = & (|W_{i,k-1}| - 1)2^{m_k} \dfrac{C_{i,k-1} - 2(a_i + 1)}{|W_{i,k-1}| - 1} + 2^{m_k} \dfrac{(2a_k)2^{m_k}}{2^{m_k}} \\ & = & 2^{m_k} C_{i,k-1} \\ & = & 2 \cdot (2^{m_k - 1}|W_{i,k-1}|) \dfrac{C_{i,k-1}}{|W_{i,k-1}|} \\ & = & 2\Big|[w]_d\Big| d(w), \end{array}$$

so the martingale process condition holds for w. Now suppose that w has a prefix in $W_{j,k-1}$ where $j \neq i$. Then $W_{j,k-1} = W_{j,k}$ and $C_{j,k-1} = C_{j,k}$, so $d(v) = d(v0) = d(v1)$ for all $v \in [w]_d$ and hence the martingale process condition holds for w. Also, the martingale process condition holds for all $w \in \{0,1\}^{<M_{k+1}-1} - \{0,1\}^{<M_k}$ because again $d(v) = d(v0) = d(v1)$ for all $v \in [w]_d$. Therefore, by induction, d is a martingale process.

For all $n \geq N$, $K(S[0..n-1]) < n - \log n$, so for every $a \geq 1$ there is some $k(a) \geq 1$ such that $x_{k(a)} \sqsubseteq S$ and $a_{k(a)} = a$. Then

$$d(S[0..M_{k(a)} - 1]) = \frac{C_{k(a),k(a)}}{|W_{k(a),k(a)}|} = 2a$$

for all $a \geq 1$, so it follows that $S \in S^\infty[d]$. The theorem now follows by Lemma 4.2. □

Corollary 4.8. *There exist a computably random sequence S and a polynomial-time non-negative martingale process d such that d succeeds on S.*

Proof. Lathrop and Lutz [8] proved that there is a computably random sequence $S \in \mathbf{C}$ that is *ultracompressible* in the sense that for every computable, non-decreasing, unbounded function $g : \mathbb{N} \to \mathbb{N}$, for all but finitely many $n \in \mathbb{N}$, $K(S[0..n-1]) < K(n) + g(n)$. Such a sequence S clearly satisfies the hypothesis of Theorem 4.7. □

Acknowledgment. The second author thanks Jennifer Chayes for asking a question that prompted this research at the September, 2001 Workshop on Computational Complexity and Statistical Physics in Santa Fe, New Mexico. We thank the referees for several helpful suggestions.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley, 1992.

[2] K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory,* Lecture Notes in Pure and Applied Mathematics, pages 1–47. Marcel Dekker, New York, N.Y., 1997.

[3] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, second edition, 1995.

[4] K. L. Chung. *A Course in Probability Theory.* Academic Press, third edition, 2001.

[5] J. L. Doob. Regularity properties of certain families of chance variables. *Transactions of the American Mathematical Society*, 47:455–486, 1940.

[6] J. L. Doob. *Stochastic Processes.* Wiley, New York, N.Y., 1953.

[7] R. Durrett. *Essentials of Stochastic Processes.* Springer, 1999.

[8] J. I. Lathrop and J. H. Lutz. Recursive computational depth. *Information and Computation*, 153(2):139–172, 1999.

[9] P. Lévy. Propriétés asymptotiques des sommes de variables indépendantes ou enchainées. *Journal des mathématiques pures et appliquées. Series 9.*, 14(4):347–402, 1935.

[10] P. Lévy. *Théorie de l'Addition des Variables Aleatoires.* Gauthier-Villars, 1937 (second edition 1954).

[11] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, Berlin, 1997. Second Edition.

[12] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.

[13] J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–254. Springer-Verlag, 1997.

[14] J. H. Lutz. Resource-bounded measure. In *Proceedings of the 13th IEEE Conference on Computational Complexity*, pages 236–248. IEEE Computer Society, 1998.

[15] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[16] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[17] S. M. Ross. *Stochastic Processes.* Wiley, 1983.

[18] C. P. Schnorr. Klassifikation der Zufallsgesetze nach Komplexität und Ordnung. *Z. Wahrscheinlichkeitstheorie verw. Geb.*, 16:1–21, 1970.

[19] C. P. Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5:246–258, 1971.

[20] C. P. Schnorr. Zufälligkeit und Wahrscheinlichkeit. *Lecture Notes in Mathematics*, 218, 1971.

[21] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences*, 7:376–388, 1973.

[22] J. Ville. *Étude Critique de la Notion de Collectif.* Gauthier–Villars, Paris, 1939.