Weakly Hard Problems[*]

Jack H. Lutz
Department of Computer Science
Iowa State University
Ames, Iowa 50011

Abstract

A weak completeness phenomenon is investigated in the complexity class $\mathrm{E} = \mathrm{DTIME}(2^{\mathrm{linear}})$. According to standard terminology, a language H is *$\leq_m^{\mathrm{P}}$-hard* for E if the set $\mathrm{P}_m(H)$, consisting of all languages $A \leq_m^{\mathrm{P}} H$, contains the entire class E. A language C is *$\leq_m^{\mathrm{P}}$-complete* for E if it is $\leq_m^{\mathrm{P}}$-hard for E and is also an element of E. Generalizing this, a language H is *weakly $\leq_m^{\mathrm{P}}$-hard* for E if the set $\mathrm{P}_m(H)$ does not have measure 0 in E. A language C is *weakly $\leq_m^{\mathrm{P}}$-complete* for E if it is weakly $\leq_m^{\mathrm{P}}$-hard for E and is also an element of E.

The main result of this paper is the construction of a language that is weakly $\leq_m^{\mathrm{P}}$-complete, but not $\leq_m^{\mathrm{P}}$-complete, for E. The existence of such languages implies that previously known strong lower bounds on the complexity of weakly $\leq_m^{\mathrm{P}}$-hard problems for E (given by work of Lutz, Mayordomo, and Juedes) are indeed more general than the corresponding bounds for $\leq_m^{\mathrm{P}}$-hard problems for E.

The proof of this result introduces a new diagonalization method, called *martingale diagonalization*. Using this method, one simultaneously develops an infinite family of polynomial time computable martingales (betting strategies) and a corresponding family of languages that defeat these martingales (prevent them from winning too much money) while also pursuing another agenda. Martingale diagonalization may be useful for a variety of applications.

[*]This research was supported in part by National Science Foundation Grant CCR-9157382, with matching funds from Rockwell International and Microware Systems Corporation.

1 Introduction

In practice to date, proving that a decision problem (i.e., language) $H \subseteq \{0,1\}^*$ is computationally intractable usually amounts to proving that every member of the complexity class $\mathrm{E} = \mathrm{DTIME}(2^{\mathrm{linear}})$—or some larger class—is efficiently reducible to H. (See [25] for a survey of such arguments.) For example, some problems involving the existence of winning strategies for certain two-person combinatorial games are known to be intractable because they are polynomial time many-one hard (in fact, logarithmic space many-one complete) for E [24].

Briefly, a language H is *polynomial time many-one hard* (abbreviated $\leq^{\mathrm{P}}_{m}$-*hard*) for E if every language $A \in \mathrm{E}$ is polynomial time many-one reducible to H (abbreviated $A \leq^{\mathrm{P}}_{m} H$). A language C is $\leq^{\mathrm{P}}_{m}$-*complete* for E if $C \in \mathrm{E}$ and C is $\leq^{\mathrm{P}}_{m}$-hard for E.

A language H that is $\leq^{\mathrm{P}}_{m}$-hard for E is clearly intractable in the sense that $H \notin \mathrm{P}$, i.e., H is not decidable in polynomial time. This is because a well-known diagonalization argument [3] shows that there is a language $B \in \mathrm{E} - \mathrm{P}$. Since $B \in \mathrm{E}$, it must be the case that $B \leq^{\mathrm{P}}_{m} H$. Since $B \notin \mathrm{P}$, it follows that $H \notin \mathrm{P}$.

In fact, languages that are $\leq^{\mathrm{P}}_{m}$-hard for E are known to have much stronger intractability properties. Three examples follow.

(A) Meyer [15] has shown that every $\leq^{\mathrm{P}}_{m}$-hard language H for E is *dense*. This means that there is a real number $\varepsilon > 0$ such that, for all sufficiently large n, H contains at least $2^{n^{\varepsilon}}$ strings $x \in \{0,1\}^{\leq n}$.

(B) Schöning [23] and Huynh [6] have shown that every $\leq^{\mathrm{P}}_{m}$-hard language H for E is hard to approximate in the sense that, for every language $A \in \mathrm{P}$, the symmetric difference $A \bigtriangleup H$ is dense. (Note that this immediately implies result (A) above.)

(C) Orponen and Schöning [16] have shown that every $\leq^{\mathrm{P}}_{m}$-hard language H for E has a dense polynomial complexity core K. This condition, defined precisely in section 2 below, means roughly that K is dense and that every Turing machine that is consistent with H performs badly (either by running for more than polynomially many steps or by failing to decide) on all but finitely many inputs $x \in K$.

In fact, the proofs of results (A), (B), and (C) all have the same overall structure as the proof that no $\leq_m^{\mathrm{P}}$-hard language H for E is in P. In each case, a "very intractable" language $B \in \mathrm{E}$ is exhibited by diagonalization. This intractability of B, together with the fact that $B \leq_m^{\mathrm{P}} H$, is then shown to imply the appropriate intractability property for H.

At this time, it appears likely that most interesting intractable problems are not $\leq_m^{\mathrm{P}}$-hard for E or larger classes. Insofar as this is true, results such as (A), (B), and (C) above fail to have interesting cases. Lutz [9] proposed to remedy this limitation by weakening the requirement that H be $\leq_m^{\mathrm{P}}$-hard for E in such results.

To be more specific, given a language H, the $\leq_m^{\mathrm{P}}$*-span* of H (also called the *lower* $\leq_m^{\mathrm{P}}$*-span* of H [7]) is the set

$$\mathrm{P}_m(H) = \left\{ A \subseteq \{0,1\}^* \,\middle|\, A \leq_m^{\mathrm{P}} H \right\},$$

consisting of all languages that are polynomial time many-one reducible to H. The language H is $\leq_m^{\mathrm{P}}$-hard for E if $\mathrm{E} \subseteq \mathrm{P}_m(H)$, i.e., if $\mathrm{P}_m(H)$ contains *all* of the complexity class E. Lutz [9] proposed consideration of weaker hypotheses, stating only that $\mathrm{P}_m(H)$ contains a non-negligible subset of E.

The expression "non-negligible subset of E" can be assigned two useful meanings, one in terms of *resource-bounded category* [9] and the other in terms of *resource-bounded measure* [10, 8]. (*Caution:* Resource-bounded measure was incorrectly formulated in [9]. The present paper refers only to the corrected formulation, in terms of martingales, presented in [10, 8] and discussed briefly in section 3 below.) Resource-bounded category, a complexity-theoretic generalization of classical Baire category [17], led to an extension of result (B) above in [9]. Work since [9] has focused instead on resource-bounded measure.

Resource-bounded measure is a generalization of classical Lebesgue measure [2, 18, 17]. As such, it has Lebesgue measure as a special case, but other special cases provide internal measures for various complexity classes. This paper concerns the special case of measure in the complexity class E. In particular, resource-bounded measure defines precisely what it means for a set X of languages to have *measure 0 in* E. This condition, written $\mu(X \mid \mathrm{E}) = 0$, means intuitively that $X \cap \mathrm{E}$ is a *negligibly small* subset of E. (This intuition is justified technically in [10] and in section 3 below.) A set Y of languages has *measure 1 in* E, written $\mu(Y \mid \mathrm{E}) = 1$, if $\mu(Y^c \mid \mathrm{E}) = 0$, where Y^c is the complement of Y. In this latter case, Y is said to contain *almost every* language in E.

It is emphasized here that not every set X of languages has a measure ("is measurable") in E. In particular, the expression "$\mu(X \mid \mathrm{E}) \neq 0$" only means that X does not have measure 0 in E. It does not necessarily imply that X has some other measure in E.

Generalizing the notion of $\leq_m^{\mathrm{P}}$-hardness for E, say that a language H is *weakly $\leq_m^{\mathrm{P}}$-hard* for E if $\mu(\mathrm{P}_m(H) \mid \mathrm{E}) \neq 0$, i.e., if $\mathrm{P}_m(H)$ does not have measure 0 in E. Similarly, say that a language C is *weakly $\leq_m^{\mathrm{P}}$-complete* for E if $C \in \mathrm{E}$ and C is weakly $\leq_m^{\mathrm{P}}$-hard for E. Since E does not have measure 0 in E [10], it is clear that every $\leq_m^{\mathrm{P}}$-hard language for E is weakly $\leq_m^{\mathrm{P}}$-hard for E, and hence that every $\leq_m^{\mathrm{P}}$-complete language for E is weakly $\leq_m^{\mathrm{P}}$-complete for E.

The following extensions of results (A), (B), and (C) above are now known.

(A′) Lutz and Mayordomo [12] have shown that every weakly $\leq_m^{\mathrm{P}}$-hard language H for E (in fact, every $\leq_{n^\alpha-tt}^{\mathrm{P}}$-hard language for E, for $\alpha < 1$) is dense.

(B′) The method of [12] extends in a straightforward matter to show that, for every weakly $\leq_m^{\mathrm{P}}$-hard language H for E and every language $A \in \mathrm{P}$, the symmetric difference $A \bigtriangleup H$ is dense.

(C′) Juedes and Lutz [7] have shown that every weakly $\leq_m^{\mathrm{P}}$-hard language H for E has a dense exponential complexity core K. (This condition, defined in section 2, implies immediately that K is a dense polynomial complexity core of H.)

Results (A′), (B′), and (C′) extend the strong intractability results (A), (B), and (C) from $\leq_m^{\mathrm{P}}$-hard languages for E to weakly $\leq_m^{\mathrm{P}}$-hard languages for E. This extends the class of problems to which well-understood lower bound techniques can be applied, *unless every weakly $\leq_m^{\mathrm{P}}$-hard language for* E *is already $\leq_m^{\mathrm{P}}$-hard for* E. Surprisingly, although weak $\leq_m^{\mathrm{P}}$-hardness appears to be a weaker hypothesis than $\leq_m^{\mathrm{P}}$-hardness, this has not been proven to date.

The present paper remedies this situation. In fact, the Main Theorem, in section 4 below, says that there exist languages that are weakly $\leq_m^{\mathrm{P}}$-complete, but not $\leq_m^{\mathrm{P}}$-complete, for E. It follows that results (A′), (B′), and (C′) do indeed extend the class of problems for which strong intractability results can be proven.

The Main Theorem is proven by means of a new diagonalization method, called *martingale diagonalization*. This method involves the simultaneous construction, by a mutual recursion, of (i) an infinite sequence of polynomial time computable martingales (betting strategies); and (ii) a corresponding sequence of languages that defeat these martingales (prevent them from winning too much money), while also pursuing another agenda. The interplay between these two constructions ensures that the sequence of languages in (ii) can be used to construct a language that is weakly $\leq_m^{\mathrm{P}}$-complete, but not $\leq_m^{\mathrm{P}}$-complete for E. Martingale diagonalization may turn out to be useful for a variety of applications.

The proof of the Main Theorem also makes essential use of a recent theorem of Juedes and Lutz [7], which gives a nontrivial *upper* bound on the complexities of $\leq_m^{\mathrm{P}}$-hard languages for E.

Section 2 presents basic notation and definitions. Section 3 provides definitions and basic properties of feasible (polynomial time computable) martingales, uses these to define measure in E, and proves a new result, the Rigid Enumeration Theorem. This result provides a uniform enumeration of feasible martingales that is crucial for the martingale diagonalization method. Section 4 is devoted entirely to the Main Theorem and its proof. Section 5 briefly discusses directions for future work, with particular emphasis on the search for *natural* problems that are weakly $\leq_m^{\mathrm{P}}$-hard for E.

2 Preliminaries

All *languages* (synonymously, *decision problems*) in this paper are sets of binary strings, i.e., sets $A \subseteq \{0,1\}^*$.

The *standard enumeration* of $\{0,1\}^*$ is the infinite sequence

$$\lambda, 0, 1, 00, 01, 10, 11, 000, 001, \cdots$$

in which strings appear first in order of length, then in lexicographic order. The symbol λ denotes the *empty string* and the expression $|w|$ denotes the *length* of a string $w \in \{0,1\}^*$. It is convenient to write the standard enumeration in the form

$$\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \cdots.$$

That is, for each $n \in \mathbf{N}$, $\mathbf{n}$ is the n^{th} string (counting from 0) in the standard enumeration of $\{0,1\}^*$. Thus, $\mathbf{0} = \lambda$, $\mathbf{1} = 0$, $\mathbf{2} = 1$, $\mathbf{3} = 00$, etc. Note also that $|\mathbf{n}|$ denotes the length of the n^{th} string in $\{0,1\}^*$.

The *Boolean value* of a condition ψ is

$$[\![\psi]\!] = \begin{cases} 1 & \text{if } \psi \text{ is true} \\ 0 & \text{if } \psi \text{ is false.} \end{cases}$$

Each language $A \subseteq \{0,1\}^*$ is identified with its *characteristic sequence*, which is the infinite binary sequence

$$\chi_A = [\![\mathbf{0} \in A]\!][\![\mathbf{1} \in A]\!][\![\mathbf{2} \in A]\!] \cdots.$$

The expression $\chi_A[0..n-1]$ denotes the string consisting of the first n bits of χ_A.

This paper uses the standard *pairing function*

$$\langle , \rangle : \mathbf{N} \times \mathbf{N} \xrightarrow[onto]{1-1} \mathbf{N}$$

defined by

$$\langle k, n \rangle = \binom{k+n+1}{2} + k$$

for all $k, n \in \mathbf{N}$. This pairing function induces the pairing function

$$\langle , \rangle : \{0,1\}^* \times \{0,1\}^* \xrightarrow[onto]{1-1} \{0,1\}^*$$

defined in the obvious way, i.e., $\langle \mathbf{k}, \mathbf{n} \rangle$ is the $\langle k, n \rangle^{\text{th}}$ string in the standard enumeration of $\{0,1\}^*$. Note that $|\langle \mathbf{k}, \mathbf{n} \rangle| \leq 2(|\mathbf{k}| + |\mathbf{n}|)$ for all $\mathbf{k}, \mathbf{n} \in \{0,1\}^*$.

As noted in section 1, a language $A \subseteq \{0,1\}^*$ is *dense* if there is a real number $\varepsilon > 0$ such that, for all sufficiently large n, A contains at least 2^{n^ε} strings $x \in \{0,1\}^{\leq n}$.

Given a function $t : \mathbf{N} \to \mathbf{N}$, the complexity class $\mathrm{DTIME}(t(n))$ consists of every language $A \subseteq \{0,1\}^*$ such that $[\![x \in A]\!]$ is computable (by a deterministic Turing machine) in $O(t(|x|))$ steps. Similarly, the complexity class $\mathrm{DTIMEF}(t(n))$ consists of every function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f(x)$ is computable in $O(t(|x|))$ steps. The complexity classes

$$\mathrm{P} \quad = \quad \bigcup_{k=0}^{\infty} \mathrm{DTIME}(n^k),$$

$$\begin{aligned} \mathrm{PF} &= \bigcup_{k=0}^{\infty} \mathrm{DTIMEF}(n^k), \\ \mathrm{E} &= \bigcup_{k=0}^{\infty} \mathrm{DTIME}(2^{kn}), \\ \mathrm{E}_2 &= \bigcup_{k=0}^{\infty} \mathrm{DTIME}(2^{n^k}) \end{aligned}$$

are of particular interest in this paper.

A language A is *polynomial time many-one reducible* to a language B, written $A \leq_m^{\mathrm{P}} B$, if there is a function $f \in \mathrm{PF}$ such that $A = f^{-1}(B)$, i.e., for all $x \in \{0,1\}^*$, $x \in A \iff f(x) \in B$.

Complexity cores, first introduced by Lynch [13], have been studied extensively. The rest of this section specifies the notions of complexity cores mentioned in section 1.

Given a (deterministic Turing) machine M and an input $x \in \{0,1\}^*$, write

$$M(x) = \begin{cases} 1 & \text{if } M \text{ accepts } x \\ 0 & \text{if } M \text{ rejects } x \\ \perp & \text{in any other case.} \end{cases}$$

If $M(x) \in \{0,1\}$, then $\mathrm{time}_M(x)$ denotes the number of steps used in the computation of $M(x)$. If $M(x) = \perp$, then $\mathrm{time}_M(x) = \infty$. A machine M is *consistent* with a language A if $M(x) = [\![x \in A]\!]$ whenever $M(x) \in \{0,1\}$.

Definition. Let $t : \mathbf{N} \to \mathbf{N}$ be a time bound and let $A, K \subseteq \{0,1\}^*$. Then K is a DTIME($t(n)$)-*complexity core* of A if, for every $c \in \mathbf{N}$ and every machine M that is consistent with A, the "fast set"

$$F = \{\, x \mid \mathrm{time}_M(x) \leq c \cdot t(|x|) + c \}$$

has finite intersection with K. (By the definition of $\mathrm{time}_M(x)$, $M(x) \in \{0,1\}$ for all $x \in F$. Thus F is the set of all strings that M "decides efficiently.")

Note that every subset of a DTIME($t(n)$)-complexity core of A is a DTIME($t(n)$)-complexity core of A. Note also that, if $s(n) = O(t(n))$, then every DTIME($t(n)$)-complexity core of A is a DTIME($s(n)$)-complexity core of A.

Definition. Let $A, K \subseteq \{0,1\}^*$.

1. K is a *polynomial complexity core* of A if K is a DTIME(n^k)-complexity core of A for all $k \in \mathbf{N}$.

2. K is an *exponential complexity core* of A if there is a real number $\epsilon > 0$ such that K is a DTIME(2^{n^ϵ})-complexity core of A.

Intuitively, a P-complexity core of A is a set of infeasible instances of A, while an exponential complexity core of A is a set of extremely hard instances of A.

3 Feasible Martingales

This section presents some basic properties of martingales (betting strategies) that are computable in polynomial time. Such martingales are used to develop a fragment of resource-bounded measure that is sufficient for understanding the notion of weakly hard problems. This section also proves the Rigid Enumeration Theorem, which is crucial for the martingale diagonalization method used to prove the Main Theorem in section 4.

Definition. A *martingale* is a function $d : \{0,1\}^* \rightarrow [0,\infty)$ with the property that, for all $w \in \{0,1\}^*$,

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

A martingale d *succeeds* on a language $A \subseteq \{0,1\}^*$ if

$$\limsup_{n\to\infty} d(\chi_A[0..n-1]) = \infty.$$

(Recall that $\chi_A[0..n-1]$ is the string consisting of the first n bits of the characteristic sequence of A.) Finally, for each martingale d, define the set

$$S^\infty[d] = \{\, A \subseteq \{0,1\}^* \mid d \text{ succeeds on } A \,\}.$$

Intuitively, a martingale d is a betting strategy that, given a language A, starts with capital (amount of money) $d(\lambda)$ and bets on the membership or nonmembership of the successive strings $\mathbf{0}, \mathbf{1}, \mathbf{2}, \cdots$ (the standard enumeration of $\{0,1\}^*$) in A. Prior to betting on a string $\mathbf{n}$, the strategy has capital $d(w)$, where

$$w = [\![\mathbf{0} \in A]\!] \cdots [\![\mathbf{n-1} \in A]\!].$$

After betting on the string $\mathbf{n}$, the strategy has capital $d(wb)$, where $b = [\![\mathbf{n} \in A]\!]$. Condition (3.1) ensures that the betting is fair. The strategy succeeds on A if its capital is unbounded as the betting progresses.

Example 3.1. Define $d : \{0,1\}^* \to [0,\infty)$ by the following recursion. Let $w \in \{0,1\}^*$ and $b \in \{0,1\}$.

(i) $d(\lambda) = 1$.

(ii) $d(wb) = 2 \cdot d(w) \cdot [\![b = [\![\, |w| \text{ is prime}]\!]]\!]$.

(See Figure 1.) It is easily checked that d is a martingale that succeeds on the language $A = \{\, \mathbf{p} \,|\, p \text{ is prime}\}$ and on no other language.

Example 3.2. Define $d : \{0,1\}^* \to [0,\infty)$ by the following recursion. Let $w \in \{0,1\}^*$.

(i) $d(\lambda) = 1$.

(ii) $d(w0) = \frac{3}{2}d(w)$.

(iii) $d(w1) = \frac{1}{2}d(w)$.

(See Figure 2.) It is obvious that d is a martingale that succeeds on every finite language A. In fact, it is easily checked that $S^\infty[d]$ contains exactly every language A for which the quantity

$$\#(0, \chi_A[0..n-1]) - \frac{n}{\log 3}$$

is unbounded as $n \to \infty$, where $\#(0,w)$ denotes the number of 0's in the string w.

Martingales were used extensively by Schnorr [19, 20, 21, 22] in his investigation of random and pseudorandom sequences. Lutz [10, 8] used martingales that are computable in polynomial time to characterize sets that have measure 0 in E.

Since martingales are real-valued, their computations must employ finite approximations of real numbers. For this purpose, let

$$\mathbf{D} = \left\{ m \cdot 2^{-n} \;\middle|\; m, n \in \mathbf{N} \right\}$$

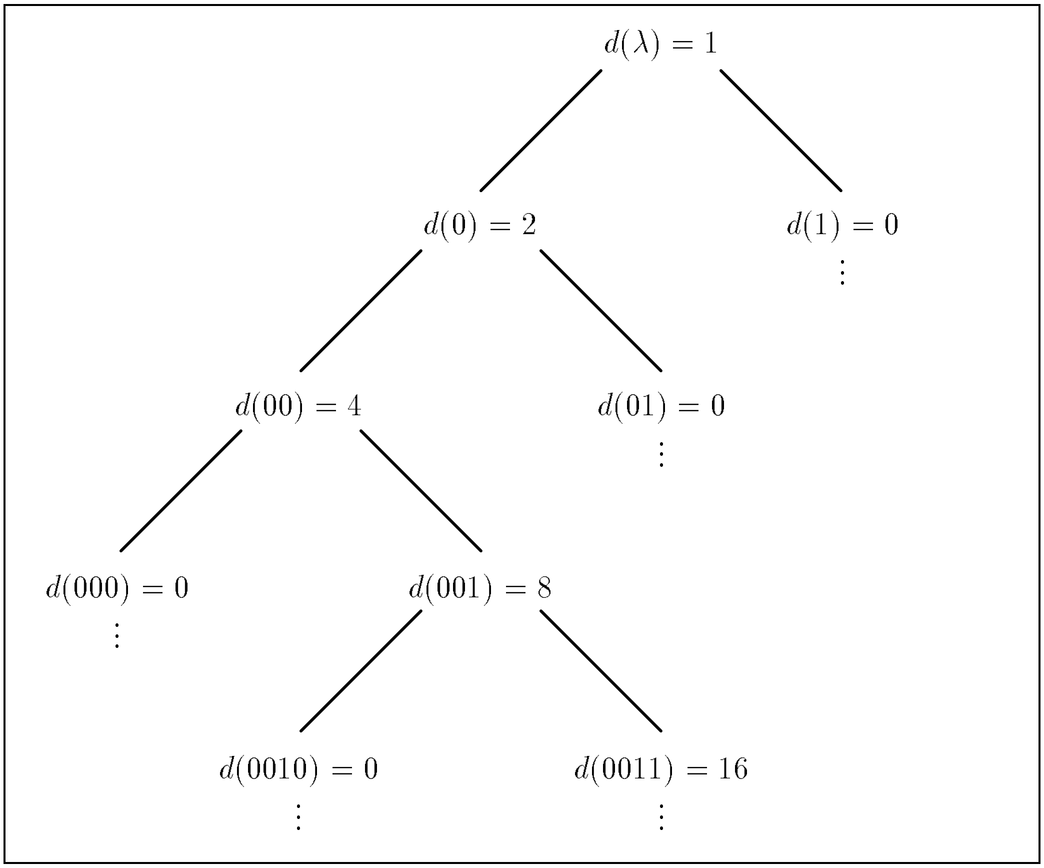


Figure 1: The martingale d of Example 3.1

be the set of *nonnegative dyadic rationals.* These are nonnegative rational numbers with finite binary expansions.

Definition. 1. A *computation* of a martingale d is a function $\widehat{d} : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ such that

$$\left|\widehat{d}_r(w) - d(w)\right| \leq 2^{-r} \tag{3.2}$$

for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$ satisfying $r \geq |w|$, where $\widehat{d}_r(w) = \widehat{d}(r, w)$.

2. A *strong computation* of a martingale d is a computation $\widehat{d}$ of d that satisfies (3.2) for *all* $r \in \mathbf{N}$ and $w \in \{0,1\}^*$.

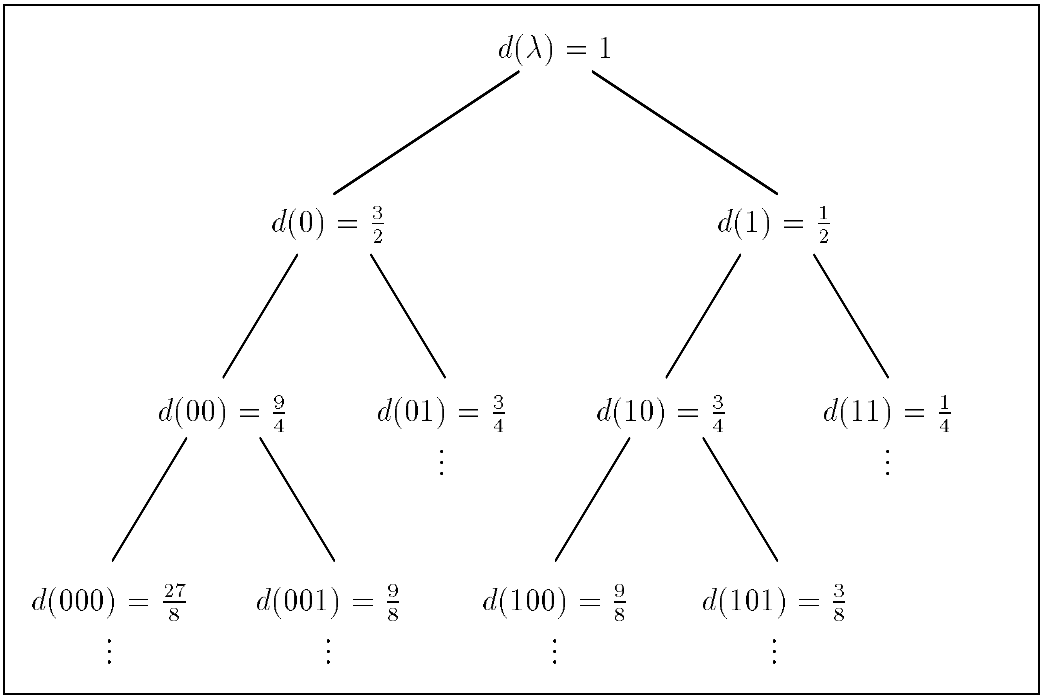


Figure 2: The martingale d of Example 3.2

3. A computation $\widehat{d}$ of a martingale d is *rigid* if it has the following two properties.

(a) For each $r \in \mathbf{N}$, the function $\widehat{d}_r$ is a martingale.

(b) For all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$, if $r \geq |w|$, then

$$\left|\widehat{d}_r(w) - \widehat{d}_{r+1}(w)\right| \leq 2^{-(r+1)}.$$

4. A p-*computation* of a martingale d is a computation $\widehat{d}$ of d such that $\widehat{d}_r(w)$ is computable in time polynomial in $r + |w|$.

5. A p-*martingale* is a martingale that has a p-computation.

A martingale is here considered to be "feasible" if and only if it is a p-martingale, i.e., if and only if it has a p-computation. Intuitively, one might prefer to insist that "feasible" martingales have *strong* p-computations, thereby avoiding the *ad hoc* condition $r \geq |w|$. On the other hand, in the technical arguments of this paper, it is useful to have *rigid* p-computations, for reasons explained below. Fortunately, the following lemma shows that all three of these conditions are equivalent.

Lemma 3.3 (Rigid Computation Lemma). For a martingale d, the following three conditions are equivalent.

(1) d has a p-computation.

(2) d has a strong p-computation.

(3) d has a rigid p-computation.

Proof. It is trivial that (3) implies (1). To see that (1) implies (2), let $\widehat{d}$ be a p-computation of d. Then the function $\widetilde{d} : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ defined by $\widetilde{d}_r(w) = \widehat{d}_{r+|w|}(w)$ is easily seen to be a strong p-computation of d, so (2) holds.

To see that (2) implies (3), let $\widehat{d}$ be a strong p-computation of d. Define a function $\widetilde{d} : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ by the following recursion. Assume that $r \in \mathbf{N}$, $w \in \{0,1\}^*$, $b \in \{0,1\}$, and $\overline{b} = 1 - b$.

(i) $\widetilde{d}_r(\lambda) = \widehat{d}_{2r+2}(\lambda)$.

(ii) $\widetilde{d}_r(wb) = \widetilde{d}_r(w) + \frac{\widehat{d}_{2r+2}(wb) - \widehat{d}_{2r+2}(w\overline{b})}{2}$.

It suffices to show that $\widetilde{d}$ is a rigid p-computation of d.

It is first shown, by induction on w, that

$$\left|\widetilde{d}_r(w) - d(w)\right| \leq 2^{-(2r+2)}(1 + |w|) \tag{3.3}$$

holds for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$. For $w = \lambda$, this follows immediately from the facts that $\widetilde{d}_r(\lambda) = \widehat{d}_{2r+2}(\lambda)$ and $\widehat{d}$ is a p-computation of d. For the induction step, assume that (3.3) holds. Then, for $b \in \{0,1\}$,

$$\left|\widetilde{d}_r(wb) - d(wb)\right| = \left|\widetilde{d}_r(w) + \frac{\widehat{d}_{2r+2}(wb) - \widehat{d}_{2r+2}(w\overline{b})}{2} - d(wb)\right|$$

$$
\begin{aligned}
&\leq \left|\tilde{d}_r(w) - d(w)\right| + \left|d(w) + \frac{\hat{d}_{2r+2}(wb) - \hat{d}_{2r+2}(w\overline{b})}{2} - d(wb)\right| \\
&= \left|\tilde{d}_r(w) - d(w)\right| + \left|\frac{d(wb) + d(w\overline{b})}{2} + \frac{\hat{d}_{2r+2}(wb) - \hat{d}_{2r+2}(w\overline{b})}{2} - d(wb)\right| \\
&= \left|\tilde{d}_r(w) - d(w)\right| + \left|\frac{\hat{d}_{2r+2}(wb) - d(wb)}{2} + \frac{d(w\overline{b}) - \hat{d}_{2r+2}(w\overline{b})}{2}\right| \\
&\leq \left|\tilde{d}_r(w) - d(w)\right| + \frac{1}{2}\left|\hat{d}_{2r+2}(wb) - d(wb)\right| + \frac{1}{2}\left|\hat{d}_{2r+2}(w\overline{b}) - d(w\overline{b})\right| \\
&\leq 2^{-(2r+2)}(1 + |w|) + 2^{-(2r+2)} \\
&= 2^{-(2r+2)}(1 + |wb|).
\end{aligned}
$$

(The last inequality holds by the induction hypothesis and the fact that $\hat{d}$ is a strong p-computation of d.) This confirms that (3.3) holds for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$.

Now let $r \in \mathbf{N}$ and $w \in \{0,1\}^*$ be such that $r \geq |w|$. Then, by (3.3),

$$
\begin{aligned}
\left|\tilde{d}_r(w) - d(w)\right| &\leq 2^{-(2r+2)}(1 + |w|) \\
&\leq 2^{-(2r+2)}(1 + r) \\
&\leq 2^{-(r+2)}.
\end{aligned}
\tag{3.4}
$$

This shows that $\tilde{d}$ is a computation of d. In fact, since $\hat{d}$ is a p-computation, it is easily checked that $\tilde{d}$ is a p-computation of d. The fact that $\tilde{d}$ is rigid follows from the following two observations.

(a) For each $r \in \mathbf{N}$, the function $\tilde{d}_r$ is clearly a martingale by clause (ii) in the definition of $\tilde{d}$.

(b) For all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$, by (3.4),

$$
\begin{aligned}
\left|\tilde{d}_r(w) - \tilde{d}_{r+1}(w)\right| &\leq \left|\tilde{d}_r(w) - d(w)\right| + \left|\tilde{d}_{r+1}(w) - d(w)\right| \\
&\leq 2^{-(r+2)} + 2^{-(r+3)} \\
&< 2^{-(r+1)}.
\end{aligned}
$$

Thus (3) holds. □

Note that the above proof does not construct a p-computation of d that is both strong and rigid. In fact, it seems reasonable to conjecture that there exists a p-martingale d for which no p-computation is both strong and rigid.

Note that a function $\widehat{d} : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ is a rigid computation of *some* martingale d if and only if it satisfies the predicates

$$\alpha_{r,w}(\widehat{d}) \equiv \left[r < |w| \text{ or } \left| \widehat{d}_r(w) - \widehat{d}_{r+1}(w) \right| \leq 2^{-(r+1)} \right]$$

and

$$\beta_{r,w}(\widehat{d}) \equiv \left[\widehat{d}_r(w) = \frac{\widehat{d}_r(w0) + \widehat{d}_r(w1)}{2} \right]$$

for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$. The next theorem exploits this fact to give a very useful enumeration of all p-martingales. The following definition specifies the useful properties of this enumeration.

Definition. A *rigid enumeration* $d_0, d_1, \cdots ; \widehat{d}_0, \widehat{d}_1, \cdots$ of all p-martingales consists of a sequence $d_0, d_1, \cdots$ and a sequence $\widehat{d}_0, \widehat{d}_1, \cdots$ with the following properties.

(i) $d_0, d_1, \cdots$ is an enumeration of all p-martingales.

(ii) For each $k \in \mathbf{N}$, $\widehat{d}_k$ is a rigid p-computation of d_k.

(iii) There is an algorithm that, given $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$, computes $\widehat{d}_{k,r}(w)$ in at most $(2 + r + |w|)^{|\mathbf{k}|}$ steps.

The following theorem is the main result of this section.

Theorem 3.4 (Rigid Enumeration Theorem). There exists a rigid enumeration of all p-martingales.

Proof. Fix a function $\widetilde{g} : \mathbf{N}^2 \times \{0,1\}^* \to \mathbf{D}$ with the following properties. (Write $\widetilde{g}_{k,r}(w) = \widetilde{g}_k(r,w) = \widetilde{g}(k,r,w)$.)

(i) $\widetilde{g}_0, \widetilde{g}_1, \cdots$ is an enumeration of all functions $f : \mathbf{N} \times \{0,1\}^* \to \mathbf{D}$ such that $f(r,w)$ is computable in time polynomial in $r + |w|$.

(ii) There is an algorithm that, given $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$, computes $\widetilde{g}_{k,r}(w)$ in at most $(2 + r + |w|)^{|\mathbf{k}|}$ steps.

(The existence of such an efficient universal function is well-known [3, 4].)

Most of this proof is devoted to two claims and their respective proofs. The first of these claims is the following.

Claim 1. There is a function $\widehat{g} : \mathbf{N}^2 \times \{0,1\}^* \to \mathbf{D}$ with the following properties. (Write $\widehat{g}_{k,r}(w) = \widehat{g}_k(r,w) = \widehat{g}(k,r,w)$.)

(a) For each $k \in \mathbf{N}$, $\widehat{g}_k$ is a rigid p-computation of some martingale g_k.

(b) For each $k \in \mathbf{N}$, if $\widetilde{g}_k$ is already a rigid p-computation of some martingale g_k, then $\widehat{g}_k = \widetilde{g}_k$.

(c) There is a constant $c \in \mathbf{N}$ such that, for all $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$, $\widehat{g}_{k,r}(w)$ is computable in at most $(2 + r + |w|)^{c+c\cdot|\mathbf{k}|}$ steps.

Assume for the moment that Claim 1 is true. Define functions $\widehat{d} : \mathbf{N}^2 \times \{0,1\}^* \to \mathbf{D}$ and $d : \mathbf{N} \times \{0,1\}^* \to [0,\infty)$ by

$$\widehat{d}_{k,r}(w) = \begin{cases} \widehat{g}_{j,r}(w) & \text{if } \mathbf{k} = 0^{c\cdot(1+|\mathbf{j}|)}1\mathbf{j} \\ 0 & \text{if } \mathbf{k} \text{ is not of this form,} \end{cases}$$

$$d_k(w) = \lim_{r\to\infty} \widehat{d}_{k,r}(w).$$

The second claim is the following.

Claim 2. The sequences $d_0, d_1, \cdots$ and $\widehat{d}_0, \widehat{d}_1, \cdots$ constitute a rigid enumeration of all p-martingales.

To prove Claim 2 (still assuming Claim 1), first note that, for all $k \in \mathbf{N}$ and $w \in \{0,1\}^*$,

$$d_k(w) = \begin{cases} g_j(w) & \text{if } \mathbf{k} = 0^{c\cdot(1+|\mathbf{j}|)}1\mathbf{j} \\ 0 & \text{if } \mathbf{k} \text{ is not of this form.} \end{cases}$$

By part (a) of Claim 1, this immediately implies that each d_k is a p-martingale. Conversely, assume that $d' : \{0,1\}^* \to [0,\infty)$ is a p-martingale. Then, by the Rigid Computation Lemma and clause (i) in the specification of $\widetilde{g}$, there is some $j \in \mathbf{N}$ such that $\widetilde{g}_j$ is a rigid p-computation of d'. Choose $k \in \mathbf{N}$ such that $\mathbf{k} = 0^{c\cdot(1+|\mathbf{j}|)}1\mathbf{j}$. Then $\widehat{d}_k = \widehat{g}_j = \widetilde{g}_j$ by part (b) of Claim 1, so $\widehat{d}_k$ is a rigid p-computation of d', so $d_k = d'$. This shows that $d_0, d_1, \cdots$ is an enumeration of all p-martingales and that each $\widehat{d}_k$ is a rigid p-computation of d_k. For $\mathbf{k} = 0^{c\cdot(1+|\mathbf{j}|)}1\mathbf{j}$, the time $t(k,r,w)$ required to compute $\widehat{d}_{k,r}(w)$ satisfies

$$\begin{aligned} t(k,r,w) &\leq |\mathbf{k}| + (2 + r + |w|)^{c\cdot(1+|\mathbf{j}|)} \\ &\leq 2^{|\mathbf{k}|-1} + (2 + r + |w|)^{|\mathbf{k}|-1} \\ &\leq (2 + r + |w|)^{|\mathbf{k}|}. \end{aligned}$$

This proves Claim 2, and hence the theorem. All that remains, then, is to prove Claim 1.

To prove Claim 1, the values $\widehat{g}_{k,r}(w)$ are first specified for all $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$. Define the following predicates. (In these predicates, it is useful to regard $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$ as parameters and $f, \widehat{f} : \mathbf{N}^2 \times \{0,1\}^* \rightarrow \mathbf{D}$ as variables.)

$$\alpha_{k,r,w}(f, \widehat{f}) \equiv \left[r < |w| \text{ or } \left|\widehat{f}_{k,r}(w) - f_{k,r+1}(w)\right| \leq 2^{-(r+1)}\right]$$

$$\beta_{k,r,w}(f, \widehat{f}) \equiv \left[\widehat{f}_{k,r}(w) = \frac{f_{k,r}(w0) + f_{k,r}(w1)}{2}\right].$$

Define $\widehat{g} : \mathbf{N}^2 \times \{0,1\}^* \rightarrow \mathbf{D}$ by recursion on r and w as follows. Let $k, r \in \mathbf{N}$, $w \in \{0,1\}^*$, and $b \in \{0,1\}$.

$$\text{(I)} \quad \widehat{g}_{k,0}(\lambda) = \widetilde{g}_{k,0}(\lambda).$$

$$\text{(II)} \quad \widehat{g}_{k,r+1}(\lambda) = \begin{cases} \widetilde{g}_{k,r+1}(\lambda) & \text{if } \alpha_{k,r,\lambda}(\widetilde{g}, \widehat{g}) \\ \widehat{g}_{k,r}(\lambda) & \text{otherwise} \end{cases}$$

$$\text{(III)} \quad \widehat{g}_{k,0}(wb) = \begin{cases} \widetilde{g}_{k,0}(wb) & \text{if } \beta_{k,0,w}(\widetilde{g}, \widehat{g}) \\ \widehat{g}_{k,0}(w) & \text{otherwise.} \end{cases}$$

$$\text{(IV)} \quad \widehat{g}_{k,r+1}(wb) = \begin{cases} \widetilde{g}_{k,r+1}(wb) & \begin{array}{l}\text{if } \alpha_{k,r,w0}(\widetilde{g}, \widehat{g}) \text{ and} \\ \quad \alpha_{k,r,w1}(\widetilde{g}, \widehat{g}) \text{ and} \\ \quad \beta_{k,r+1,w}(\widetilde{g}, \widehat{g})\end{array} \\ \widehat{g}_{k,r}(wb) + \widehat{g}_{k,r+1}(w) - \widehat{g}_{k,r}(w) & \text{otherwise} \end{cases}$$

By condition (ii) in the choice of $\widetilde{g}$, the function $\widehat{g}$ defined by this recursion is easily seen to satisfy condition (c) of Claim 1.

To see that $\widehat{g}$ satisfies condition (a) of Claim 1, let $k \in \mathbf{N}$ be arbitrary. A routine induction on r shows that $\beta_{k,r,w}(\widehat{g}, \widehat{g})$ holds for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$. It follows easily that each $\widehat{g}_{k,r}$ is a martingale. A routine induction on w then shows that $\alpha_{k,r,w}(\widehat{g}, \widehat{g})$ holds for all $r \in \mathbf{N}$ and $w \in \{0,1\}^*$. It follows that $\widehat{g}_k$ is a rigid p-computation of the martingale g_k defined by $g_k(w) = \lim_{r\rightarrow\infty} \widehat{g}_{k,r}(w)$. Thus $\widehat{g}$ satisfies condition (a) of Claim 1.

Finally, to see that $\widehat{g}$ satisfies condition (b) of Claim 1, fix $k \in \mathbf{N}$ and assume that $\widetilde{g}_k$ is a rigid computation of some martingale g_k. Then a routine induction on r and w shows that $\widehat{g}_k = \widetilde{g}_k$. (The α and β predicates hold throughout the induction, so the "otherwise" cases are never invoked in the definition of $\widehat{g}_k$.) This completes the proof of Claim 1 and the proof of the Rigid Enumeration Theorem. □

The rest of this section briefly develops those aspects of measure in E that are used in this paper. The key ideas are in the following definition.

Definition. 1. A set X of languages has p-*measure 0*, written $\mu_{\rm p}(X) = 0$, if there is a p-martingale d such that $X \subseteq S^\infty[d]$.

2. A set X of languages has *measure 0 in* E, written $\mu(X \mid {\rm E}) = 0$, if $\mu_{\rm p}(X \cap {\rm E}) = 0$.

3. A set X of languages has *measure 1 in* E, written $\mu(X \mid {\rm E}) = 1$, if $\mu(X^c \mid {\rm E}) = 0$, where X^c is the complement of X. In this case, X is said to contain *almost every* language in E.

4. The expression $\mu(X \mid {\rm E}) \neq 0$ indicates that X does *not* have measure 0 in E. Note that this does *not* assert that "$\mu(X \mid {\rm E})$" has some nonzero value.

Thus, a set X of languages has measure 0 in E if there is a feasible martingale that succeeds on every element of X.

The following fact is obvious but useful.

Proposition 3.5. Every set X of languages satisfies the implications

$$\mu_{\rm p}(X) = 0 \Longrightarrow \mu(X \mid {\rm E}) = 0, \qquad \mu_{\rm p}(X) = 0 \Longrightarrow \Pr[A \in X] = 0,$$

where the probability $\Pr[A \in X]$ is computed according to the random experiment in which a language $A \subseteq \{0,1\}^*$ is chosen probabilistically, using an independent toss of a fair coin to decide whether each string $x \in \{0,1\}^*$ is in A.

The right-hand implication in Proposition 3.5 makes it clear that p-measure 0 sets are negligibly small. What is significant for complexity theory is that, if X has measure 0 in E, then $X \cap {\rm E}$ is negligibly small *as a subset of* E. This intuition is technically justified in [10], where it is shown that finite subsets of E have measure 0 in E, and that the sets of measure 0 in E are closed under subset, finite unions, and certain countable unions, called "p-unions." Most importantly, the following is shown.

Theorem 3.6 [10]. $\mu({\rm E} \mid {\rm E}) \neq 0$.

Combined with the above-mentioned closure properties, this result (which is a special case of the more general Measure Conservation Theorem [10]) ensures that $X \cap \mathrm{E}$ is, in a nontrivial sense, a negligibly small subset of E whenever X has measure 0 in E.

4 Weak Completeness in E

In standard terminology, a language H is $\leq_m^{\mathrm{P}}$*-hard* for a complexity class $\mathcal{C}$ if the set

$$\mathrm{P}_m(H) = \left\{ A \,\middle|\, A \leq_m^{\mathrm{P}} H \right\}$$

contains all of $\mathcal{C}$. A language C is $\leq_m^{\mathrm{P}}$*-complete* for $\mathcal{C}$ if $C \in \mathcal{C}$ and C is $\leq_m^{\mathrm{P}}$-hard for $\mathcal{C}$. The following definition generalizes these notions for the complexity class $\mathcal{C} = \mathrm{E}$.

<u>Definition.</u> A language H is *weakly* $\leq_m^{\mathrm{P}}$*-hard* for E if $\mu(\mathrm{P}_m(H) \mid \mathrm{E}) \neq 0$, i.e., the set $\mathrm{P}_m(H)$ does not have measure 0 in E. A language C is *weakly* $\leq_m^{\mathrm{P}}$*-complete* for E if $C \in \mathrm{E}$ and C is weakly $\leq_m^{\mathrm{P}}$-hard for E.

By Theorem 3.6, every $\leq_m^{\mathrm{P}}$-hard language for E is weakly $\leq_m^{\mathrm{P}}$-hard for E, whence every $\leq_m^{\mathrm{P}}$-complete language for E is weakly $\leq_m^{\mathrm{P}}$-complete for E. The following result says that the converse does *not* hold, i.e., that in E, weak $\leq_m^{\mathrm{P}}$-completeness is a proper generalization of $\leq_m^{\mathrm{P}}$-completeness.

<u>Theorem 4.1</u> (Main Theorem). There is a language C that is weakly $\leq_m^{\mathrm{P}}$-complete, but not $\leq_m^{\mathrm{P}}$-complete, for E.

The rest of this section is devoted to proving the Main Theorem.

A recent theorem of Juedes and Lutz gives a necessary condition for a language to be $\leq_m^{\mathrm{P}}$-hard for E. This condition, based on an idea of Meyer [15], plays an important role in the present proof. The key ideas are developed in the following definitions.

<u>Definition.</u> The *collision set* of a function $f : \{0,1\}^* \to \{0,1\}^*$ is

$$C_f = \{\, n \in \mathbf{N} \mid (\exists m < n)\, f(\mathbf{m}) = f(\mathbf{n}) \,\}\,.$$

A function $f : \{0,1\}^* \to \{0,1\}^*$ is *one-to-one almost everywhere* if C_f is finite.

Definition. Let $A \subseteq \{0,1\}^*$ and $t : \mathbf{N} \to \mathbf{N}$. A *many-one reduction of* A is a computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that $A = f^{-1}(f(A))$, i.e., such that, for all $x \in \{0,1\}^*$, $f(x) \in f(A)$ implies $x \in A$. A $\leq_m^{\mathrm{DTIME}(t)}$*-reduction of* A is a many-one reduction f of A such that $f \in \mathrm{DTIMEF}(t)$.

Definition. Let $A \subseteq \{0,1\}^*$ and $t : \mathbf{N} \to \mathbf{N}$. Then A is *incompressible by* $\leq_m^{\mathrm{DTIME}(t)}$*-reductions* if every $\leq_m^{\mathrm{DTIME}(t)}$-reduction of A is one-to-one almost everywhere.

Intuitively, if f is a $\leq_m^{\mathrm{DTIME}(t)}$-reduction of A and C_f is large, then f compresses many questions "$x \in A$?" to fewer questions "$f(x) \in f(A)$?" If A is incompressible by $\leq_m^{\mathrm{DTIME}(t)}$-reductions, then A is "very complex" in the sense that very little such compression can occur.

The following result is used here.

Theorem 4.2 (Juedes and Lutz [7]). No language that is $\leq_m^{\mathrm{P}}$-hard for E is incompressible by $\leq_m^{\mathrm{DTIME}(2^{4n})}$-reductions.

Since almost every language (and almost every language in E) *is* incompressible by $\leq_m^{\mathrm{DTIME}(2^{4n})}$-reductions [7], Theorem 4.2 says that the $\leq_m^{\mathrm{P}}$-hard languages are "unusually simple" in at least this one respect.

The largest part of the proof of the Main Theorem is the construction of a language $H \in \mathrm{E}_2$ with the following two properties.

(I) H is weakly $\leq_m^{\mathrm{P}}$-hard for E.

(II) H is incompressible by $\leq_m^{\mathrm{DTIME}(2^{4n})}$-reductions.

By Theorem 4.2, this language H cannot be $\leq_m^{\mathrm{P}}$-hard for E. A padding argument then gives the Main Theorem.

The language H is constructed by diagonalization. In establishing property (I), the construction uses a fixed rigid enumeration $d_0, d_1, \cdots ; \widehat{d_0}, \widehat{d_1}, \cdots$ of all p-martingales. Such a rigid enumeration exists by Theorem 3.4. In establishing property (II), the construction uses a fixed function f such that $f \in \mathrm{DTIMEF}(2^{5n})$ and f is universal for $\mathrm{DTIMEF}(2^{4n})$, in the sense that

$$\mathrm{DTIMEF}(2^{4n}) = \{\, f_i \,|\, i \in \mathbf{N} \,\},$$

where $f_i(x) = f(\langle \mathbf{i}, x \rangle)$. (The existence of such an efficient universal function is well-known [3, 4].)

In addition to the pairing function $\langle , \rangle$ mentioned in section 2, the construction of H uses the ordering $<^*$ of $\mathbf{N}^2$ defined by

$$\begin{aligned}(j,m) <^* (k,n) \iff & [(1+|\mathbf{j}|)(1+|\mathbf{m}|) < (1+|\mathbf{k}|)(1+|\mathbf{n}|) \\ & \text{or } [(1+|\mathbf{j}|)(1+|\mathbf{m}|) = (1+|\mathbf{k}|)(1+|\mathbf{n}|) \\ & \quad \text{and } \langle j,m\rangle < \langle k,n\rangle]]\end{aligned}$$

for all $j,m,k,n \in \mathbf{N}$. It is easy to check that $(\mathbf{N}^2, <^*)$ is order isomorphic to $(\mathbf{N}, <)$. For $(k,n) \in \mathbf{N}^2$, let

$$\#^*(k,n) = \left| \left\{ (j,m) \in \mathbf{N}^2 \,\middle|\, (j,m) <^* (k,n) \right\} \right|$$

be the number of $<^*$-predecessors of (k,n) in $\mathbf{N}^2$. Two important properties of $<^*$ are that

$$(j,m) <^* (k,n) \Longrightarrow (1+|\mathbf{j}|)(1+|\mathbf{m}|) \leq (1+|\mathbf{k}|)(1+|\mathbf{n}|)$$

and

$$\#^*(k,n) = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}.$$

Using the ordering $<^*$, define the *modified collision set* C_i^* of a function $f_i \in \mathrm{DTIMEF}(2^{4n})$ by

$$C_i^* = \left\{ (k,n) \in \mathbf{N}^2 \,\middle|\, (\exists (j,m) <^* (k,n))\, f_i(\langle \mathbf{j},\mathbf{m}\rangle) = f_i(\langle \mathbf{k},\mathbf{n}\rangle) \right\}.$$

Also, for $k \in \mathbf{N}$, define the k^{th} *slice* of C_i^* to be the set

$$C_{i,k}^* = \{ n \in \mathbf{N} \mid (k,n) \in C_i^* \}.$$

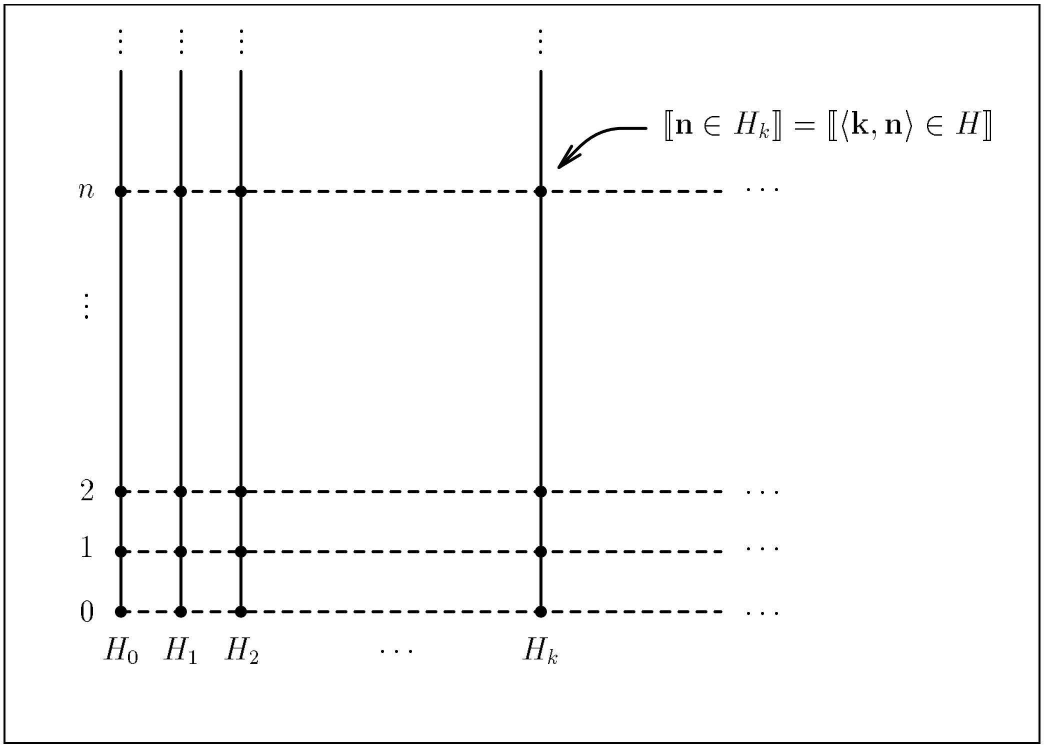
Lemma 4.3. For all $i \in \mathbf{N}$, the function f_i is one-to-one almost everywhere if and only if the set C_i^* is finite.

Proof. Fix $i \in \mathbf{N}$ and define an equivalence relation $\equiv_i$ on $\{0,1\}^*$ by

$$x \equiv_i y \iff f_i(x) = f_i(y).$$

Then the collision set C_{f_i} and the modified collision set C_i^* each consist of all but one of the elements of all the non-singleton equivalence classes of $\equiv_i$. It follows immediately that C_{f_i} and C_i^* are either both finite or both infinite. □

Overview of the Construction. Informally and intuitively, the language H is constructed by deciding the Boolean values $\langle \mathbf{k}, \mathbf{n} \rangle \in H$ for successive (k, n) in the ordering $<^*$ of $\mathbf{N}^2$. It is convenient to regard H as consisting of the separate "strands" $H_k = \{ \mathbf{n} \mid \langle \mathbf{k}, \mathbf{n} \rangle \in H \}$ for $k = 0, 1, 2, \cdots$. (See Figure 3.) The construction exploits the ordering $<^*$ to ensure that $H \in \mathrm{E}_2$ and each $H_k \in \mathrm{E}$. The "ultimate objective" of each H_k is to ensure that a specially constructed martingale $\widetilde{d}_k$ does not succeed on H_k. For each k, all but finitely many of the values $[\![\mathbf{n} \in H_k]\!]$ are chosen according to this ultimate objective. The exceptions occur when values $[\![\mathbf{n} \in H_k]\!]$ are chosen in order to "destroy" various functions $f_i \in \mathrm{DTIMEF}(2^{4n})$, i.e., in order to ensure that these functions are *not* many-one reductions of H.



$[\![\mathbf{n} \in H_k]\!] = [\![\langle \mathbf{k}, \mathbf{n} \rangle \in H]\!]$
n
2
1
0
H_0
H_1
H_2
$\cdots$
H_k

Figure 3: The strands of H.

The specially constructed martingales are of the form $\widetilde{d}_k = d_k + \sum_{i=0}^{\infty} d_{i,k}$

where d_k is taken from the rigid enumeration of all p-martingales given by Theorem 3.4 and the martingales $d_{i,k}$ are defined below. Since $\widetilde{d}_k$ does not succeed on H_k, d_k also does not succeed on H_k. Since k is arbitrary here and each $H_k \in \mathrm{P}_m(H) \cap \mathrm{E}$, it follows that $\mathrm{P}_m(H) \cap \mathrm{E}$ does not have p-measure 0, i.e., that $\mathrm{P}_m(H)$ does not have measure 0 in E. Thus H is a weakly $\leq_m^{\mathrm{P}}$-hard for E. On the other hand, since $\widetilde{d}_k$ does not succeed on H_k, none of the martingales $d_{i,k}$ succeeds on H_k. Moreover, matters are arranged so that, for every many-one reduction f_i of H with C_i^* infinite, either some $d_{i,k}$ succeeds on H_k, or else f_i is eventually "destroyed" by some value $[\![\mathbf{n} \in H_k]\!]$. It follows that H is incompressible by $\leq_m^{\mathrm{DTIME}(2^{4n})}$-reductions, whence H is not $\leq_m^{\mathrm{P}}$-hard for E by Theorem 4.2.

Precise details follow.

<u>**The Construction.**</u> The language $H \subseteq \{0,1\}^*$ is defined by

$$H = \{\, \langle \mathbf{k}, \mathbf{n} \rangle \mid \mathbf{n} \in H_k \,\},$$

where the languages $H_0, H_1, \cdots$ are defined, along with the auxiliary martingales $\widetilde{d}_0, \widetilde{d}_1, \cdots$, by the following recursion. (Recall that $d_0, d_1, \ldots; \widehat{d}_0, \widehat{d}_1, \ldots$ is a fixed rigid enumeration of all p-martingales.)

(1) For $k \in \mathbf{N}$ and $w \in \{0,1\}^*$, define

$$\widetilde{d}_k(w) = d_k(w) + \sum_{i=0}^{\infty} d_{i,k}(w),$$

where the functions $d_{i,k}$ are computed as follows. Assume that $w \in \{0,1\}^*$, $n = |w|$, and $b \in \{0,1\}$.

(a) $d_{i,k}(\lambda) = 2^{-i}$.

(b) If $(k,n) \notin C_i^*$, then $d_{i,k}(wb) = d_{i,k}(w)$.

(c) If $(k,n) \in C_i^*$, then

$$d_{i,k}(wb) = 2 \cdot d_{i,k}(w) \cdot [\![b = [\![\mathbf{m} \in H_j]\!]]\!],$$

where (j,m) is the $<^*$-least pair in $\mathbf{N}^2$ such that $f_i(\langle \mathbf{j}, \mathbf{m} \rangle) = f_i(\langle \mathbf{k}, \mathbf{n} \rangle)$.

It is clear that each $d_{i,k}$, and hence each $\widetilde{d}_k$, is a martingale.

For $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$, the approximation

$$\widehat{\widetilde{d}}_{k,r}(w) = \widehat{d}_{k,r+1}(w) + \sum_{i=0}^{r+|w|+1} d_{i,k}(w)$$

of $\widetilde{d}_k(w)$ is also used. It is easy to check that

$$\left| \widehat{\widetilde{d}}_{k,r}(w) - \widetilde{d}_k(w) \right| \leq 2^{-r}$$

for all $k, r \in \mathbf{N}$ and $w \in \{0,1\}^*$ satisfying $r + 1 \geq |w|$.

(2) In the construction of the languages $H_0, H_1, \cdots$, the operation

"destroy f_i at (k, n)"

is often performed. In all such instances, it is known that $(k, n) \in C_i^*$, and the operation is performed by setting

$$[\![\mathbf{n} \in H_k]\!] = [\![\mathbf{m} \notin H_j]\!],$$

where (j, m) is the $<^*$-least pair in $\mathbf{N}^2$ such that $f_i(\langle \mathbf{j}, \mathbf{m} \rangle) = f_i(\langle \mathbf{k}, \mathbf{n} \rangle)$. Note that a single performance of this operation ensures that f_i is not a many-one reduction of H.

The sets

$$D_{k,n} = \{\, i \in \mathbf{N} \mid (\exists (j, m) <^* (k, n))\, f_i \text{ is destroyed at } (j, m) \,\},$$

for $k, n \in \mathbf{N}$, are also used in the construction. It is emphasized that an index i appears in $D_{k,n}$ only if the operation "destroy f_i at (j, m)" is *explicitly* performed for some $(j, m) <^* (k, n)$. In particular, for each (j, m), there is at most one i such that f_i is destroyed at (j, m), even though there are many i' such that $f_i = f_{i'}$. Thus each $D_{k,n}$ is a finite set with $|D_{k,n}| \leq \#^*(k, n)$.

For $k, n \in \mathbf{N}$, let

$$\iota(k, n) = \min \{\, i \in \mathbf{N} \mid i \notin D_{k,n} \text{ and } (k, n) \in C_i^* \,\}.$$

Note that $\iota(k, n)$ is finite for all $k, n \in \mathbf{N}$ (because f_i is constant for infinitely many i). The values $[\![\mathbf{n} \in H_k]\!]$ are defined according to the following two cases.

```
begin
  $w := \chi_{H_k}[0..n-1]$;
  for $b \in \{0,1\}$ do
  begin
    $\delta_b := \widehat{d}_{k,n+1}(wb)$
    for $i := 0$ to $2n+1$ do $\delta_b := \delta_b + d_{i,k}(wb)$
  end; //Now $\delta_0 = \widehat{\widehat{d}}_{k,n}(w0)$ and $\delta_1 = \widehat{\widehat{d}}_{k,n}(w1)$.//
  if $\iota(k,n) \leq k$
     then destroy $f_{\iota(k,n)}$ at $(k,n)$
     else $\llbracket \mathbf{n} \in H_k \rrbracket := \llbracket \delta_1 \leq \delta_0 \rrbracket$
end.
```

Figure 4: Computation of $\llbracket \mathbf{n} \in H_k \rrbracket$ in the proof of Lemma 4.4.

Case 1. If $\iota(k,n) \leq k$, then destroy $f_{\iota(k,n)}$ at (k,n).

Case 2. If $\iota(k,n) > k$, then set

$$\llbracket \mathbf{n} \in H_k \rrbracket = \llbracket \widehat{\widehat{d}}_{k,n}(w1) \leq \widehat{\widehat{d}}_{k,n}(w0) \rrbracket,$$

where $w = \chi_{H_k}[0..n-1]$.

This completes the construction of the languages $H_0, H_1, \cdots$ and the martingales $\widetilde{d}_0, \widetilde{d}_1, \cdots$.

The following lemmas are used to prove the Main Theorem.

Lemma 4.4. $H \in \mathrm{E}_2$. For each $k \in \mathbf{N}$, $H_k \in \mathrm{E}$.

Proof. Assume first that $(k,n) \in \mathbf{N}^2$ and that the values $\llbracket \mathbf{m} \in H_j \rrbracket$ are known (stored) for all pairs $(j,m) <^* (k,n)$, as is the set $D_{k,n}$. Consider the computation of $\llbracket \mathbf{n} \in H_k \rrbracket$ exhibited in Figure 4.

To estimate the time required for this computation, recall the properties

$$(j, m) <^* (k, n) \Longrightarrow (1 + |\mathbf{j}|)(1 + |\mathbf{m}|) \leq (1 + |\mathbf{k}|)(1 + |\mathbf{n}|),$$

$$\#^*(k, n) = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$$

of $<^*$ and note the following.

(i) The computation of w requires at most $n \cdot \#^*(k, n) = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

(ii) The computation of $\widehat{d}_{k,n+1}(wb)$ requires at most $(3 + n + |wb|)^{|\mathbf{k}|} = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

(iii) For $0 \leq i \leq 2n+1$, the condition "$(k, n) \in C_i^*$" can be tested in at most $(1 + \#^*(k, n))^2 \cdot O(2^{5|\langle \mathbf{i}, \langle \mathbf{k}, \mathbf{n} \rangle\rangle|}) = 2^{O(|\mathbf{i}|+O((1+|\mathbf{k}|)(1+|\mathbf{n}|)))} = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

(iv) By (iii), for $0 \leq i \leq 2n + 1$, the computation of $d_{i,k}(wb)$ requires at most $O(n \cdot 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} . 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}) = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

(v) By (ii) and (iv), the entire computation of $\delta_b = \widehat{\widetilde{d}}_{k,n}(wb)$, i.e., the for-loop in Figure 4, requires at most $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} + (2n+2)2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

(vi) As in (iii), for $0 \leq i \leq k$, the condition "$(k, n) \in C_i^*$" can be tested in at most $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps. Thus, testing the condition "$\iota(k, n) \leq k$," and computing $\iota(k, n)$ if this condition is true, requires at most $(k + 1) \cdot 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps. It follows easily that the if-then-else in Figure 4 requires at most $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps.

By (i), (v), and (vi) above, the computation described in Figure 4 requires at most $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps to compute $[\![\mathbf{n} \in H_k]\!]$, given the set $D_{k,n}$ and the values $[\![\mathbf{m} \in H_j]\!]$ for $(j, m) <^* (k, n)$.

The condition $\langle \mathbf{k}, \mathbf{n} \rangle \in H$ can now be decided by computing and storing the successive values $[\![\mathbf{m} \in H_j]\!]$ according to the $<^*$-ordering of $\mathbf{N}^2$, using the computation in Figure 4 and updating $D_{j,n}$ at each stage. This requires at most $(1 + \#^*(k, n)) \cdot O(2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}) = 2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))}$ steps. Since $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} = 2^{O((1+|\langle \mathbf{k}, \mathbf{n} \rangle|)^2)}$, this proves that $H \in \mathrm{E}_2$. Also, for fixed k, $2^{O((1+|\mathbf{k}|)(1+|\mathbf{n}|))} = 2^{O(1+|\mathbf{n}|)}$, so each $H_k \in \mathrm{E}$. □

Lemma 4.5. For all $i \in \mathbf{N}$, if there exist infinitely many $k \in \mathbf{N}$ such that the slice $C^*_{i,k}$ is nonempty, then f_i is not a many-one reduction of H.

Proof. Fix $i \in \mathbf{N}$ and assume that the set

$$S = \left\{ k \in \mathbf{N} \,\middle|\, C^*_{i,k} \neq \emptyset \right\}$$

is infinite. For each $k \in S$, let $n_k = \min C^*_{i,k}$. For every $k \in S$, at least one of the following four conditions must hold.

(i) $i > k$.

(ii) $i < \iota(k, n_k)$.

(iii) $\iota(k, n_k) < i \leq k$.

(iv) $\iota(k, n_k) = i \leq k$.

(In fact, for all real numbers a, b, and c, at least one of $a > c$, $a < b$, $b < a \leq c$, $b = a \leq c$ must hold.) It is clear that condition (i) holds for only finitely many k. For each k such that condition (iii) holds, the construction of H ensures that $f_{\iota(k,n_k)}$ is destroyed at (k, n_k). Since each f_j is destroyed at most once in the construction of H, it follows that condition (iii) holds for only finitely many k. Since S is infinite, this implies that there is some $k \in S$ such that condition (ii) or condition (iv) holds.

Fix such a number k. If condition (ii) holds, then $i \in D_{k,n_k}$ (because $(k, n_k) \in C^*_i$), so f_i is not a many-one reduction of H. If condition (iv) holds, then f_i is destroyed at (k, n_k), so f_i is not a many-one reduction of H. Thus, in any case, f_i is not a many-one reduction of H. □

Lemma 4.6. For all $i, k \in \mathbf{N}$, if f_i is a many-one reduction of H and $C^*_{i,k}$ is infinite, then $d_{i,k}$ succeeds on H_k.

Proof. Assume that $i, k \in \mathbf{N}$, f_i is a many-one reduction of H, and $C^*_{i,k}$ is infinite. Consider the successive values

$$r_n = d_{i,k}(\chi_{H_k}[0..n-1])$$

for $n = 0, 1, 2 \cdots$. Clause (a) of the definition of $d_{i,k}$ says that $r_0 = 2^{-i}$, while clauses (b) and (c) ensure that each $r_{n+1} \in \{0, r_n, 2r_n\}$. In fact, since f_i is a reduction of H, clause (c) never causes r_{n+1} to be 0. We thus have the recurrence

$$r_0 = 2^{-i}, \quad r_{n+1} = \begin{cases} r_n & \text{if } n \notin C^*_{i,k} \\ 2r_n & \text{if } n \in C^*_{i,k}. \end{cases}$$

Since $C^*_{i,k}$ is infinite, this implies that $r_n \to \infty$ as $n \to \infty$, whence $d_{i,k}$ succeeds on H_k. □

Lemma 4.7. For all $k \in \mathbf{N}$, $\tilde{d}_k$ does not succeed on H_k.

Proof. Fix $k \in \mathbf{N}$ and consider the manner in which the values $[\![\mathbf{n} \in H_k]\!]$ are decided for $n = 0, 1, 2, \cdots$. There can be at most finitely many values of n for which Case 1 holds. (This is because each occurrence of Case 1 involves a new value of $\iota(k, n)$, with $\iota(k, n) \leq k$.) Thus there exists $n_0 \in \mathbf{N}$ such that Case 2 holds for all $n \geq n_0$. For all $n \in \mathbf{N}$, let

$$w_n = \chi_{H_k}[0..n-1]$$

be the n-bit prefix of χ_{H_k}. Then, for all $m \geq n_0$, Case 2 ensures that

$$\begin{aligned} \tilde{d}_k(w_{m+1}) &\leq \hat{\tilde{d}}_{k,m}(w_{m+1}) + 2^{-m} \\ &\leq \frac{\hat{\tilde{d}}_{k,m}(w_m0) + \hat{\tilde{d}}_{k,m}(w_m1)}{2} + 2^{-m} \\ &\leq \frac{\tilde{d}_k(w_m0) + 2^{-m} + \tilde{d}_k(w_m1) + 2^{-m}}{2} + 2^{-m} \\ &= \frac{\tilde{d}_k(w_m0) + \tilde{d}_k(w_m1)}{2} + 2^{1-m} \\ &= \tilde{d}_k(w_m) + 2^{1-m}. \end{aligned}$$

It follows that, for all $n \geq n_0$,

$$\tilde{d}_k(w_n) \leq \tilde{d}_k(w_{n_0}) + \sum_{m=n_0}^{n-1} 2^{1-m} < \tilde{d}_k(w_{n_0}) + 4.$$

Thus, if

$$\sigma = \max_{0 \leq n \leq n_0} \tilde{d}_k(w_n),$$

then

$$\tilde{d}_k(w_n) < \sigma + 4$$

for all $n \in \mathbf{N}$. Hence $\widetilde{d}_k$ does not succeed on H_k. □

Lemma 4.8. H is weakly $\leq^{\mathrm{P}}_m$-hard for E.

Proof. Let $k \in \mathbf{N}$. It is clear that $H_k \in \mathrm{P}_m(H)$ and $S^\infty[d_k] \subseteq S^\infty[\widetilde{d}_k]$. It follows by Lemmas 4.4 and 4.7 that

$$H_k \in \mathrm{P}_m(H) \cap \mathrm{E} - S^\infty[\widetilde{d}_k] \subseteq \mathrm{P}_m(H) \cap \mathrm{E} - S^\infty[d_k],$$

whence $\mathrm{P}_m(H) \cap \mathrm{E} \not\subseteq S^\infty[d_k]$. Since k is arbitrary here, this implies that $\mu_{\mathrm{p}}(\mathrm{P}_m(H) \cap \mathrm{E}) \neq 0$, i.e., that $\mu(\mathrm{P}_m(H) \mid \mathrm{E}) \neq 0$. Thus H is weakly $\leq^{\mathrm{P}}_m$-hard for E. □

Lemma 4.9. H is not $\leq^{\mathrm{P}}_m$-hard for E.

Proof. By Theorem 4.2, it suffices to show that H is incompressible by $\leq^{\mathrm{DTIME}(2^{4n})}_m$-reductions. For this, fix $i \in \mathbf{N}$ such that f_i is a many-one reduction of H. It suffices to show that f_i is one-to-one almost everywhere.

Note the following two things.

(i) For each $k \in \mathbf{N}$, the slice $C^*_{i,k}$ is finite by Lemmas 4.7 and 4.6.

(ii) By Lemma 4.5, there are only finitely many $k \in \mathbf{N}$ such that $C^*_{i,k} \neq \emptyset$.

Taken together, (i) and (ii) imply that C^*_i is finite. It follows by Lemma 4.3 that f_i is one-to-one almost everywhere. □

By Lemmas 4.4, 4.8, and 4.9, the language $H \in \mathrm{E}_2$ is weakly $\leq^{\mathrm{P}}_m$-hard, but not $\leq^{\mathrm{P}}_m$-hard, for E. From this, a simple padding argument suffices to prove the Main Theorem.

Proof of Main Theorem. Let H be defined as above. By Lemma 4.4, there is a polynomial $q(n) \geq n$ such that $H \in \mathrm{DTIME}(2^{q(n)})$. Let

$$C = \left\{ x10^{q(|x|)} \,\middle|\, x \in H \right\}.$$

It is easy to check that $C \in \mathrm{E}$ and that $\mathrm{P}_m(C) = \mathrm{P}_m(H)$. It follows by Lemmas 4.8 and 4.9 that C is weakly $\leq^{\mathrm{P}}_m$-complete, but not $\leq^{\mathrm{P}}_m$-complete, for E. □

5 Conclusion

The most important problem suggested by this work is to find "natural" examples of languages that are weakly $\leq^{\mathrm{P}}_{m}$-complete, but not $\leq^{\mathrm{P}}_{m}$-complete, for E. As noted in section 1, such languages would provably be strongly intractable. It is reasonable to hope that the study of such natural examples would yield new insights into the nature of intractability.

It is especially intriguing to consider the possibility that SAT and other natural NP-complete problems may be weakly $\leq^{\mathrm{P}}_{m}$-complete for E, i.e., that NP may not have measure 0 in E. The hypothesis that SAT is weakly $\leq^{\mathrm{P}}_{m}$-complete for E implies, but may in some sense be stronger than, the P $\neq$ NP hypothesis. For example, recent work has shown that, if SAT is weakly $\leq^{\mathrm{P}}_{m}$-complete for E, then NP contains P-bi-immune languages [14], every $\leq^{\mathrm{P}}_{m}$-hard language for NP is dense [12], every $\leq^{\mathrm{P}}_{m}$-complete language for NP has a dense exponential complexity core [7], and there is a language that is $\leq^{\mathrm{P}}_{\mathrm{T}}$-complete, but not $\leq^{\mathrm{P}}_{m}$-complete, for NP [11]. Further investigation of the consequences and reasonableness of this hypothesis is indicated.

It is routine to modify the proof of the Main Theorem to construct languages that are weakly $\leq^{\mathrm{P}}_{m}$-complete, but not $\leq^{\mathrm{P}}_{m}$-complete, for larger classes such as E_2 and ESPACE. A more interesting, and perhaps harder, question concerns alternate versions of the Main Theorem in which $\leq^{\mathrm{P}}_{m}$ is replaced by other reducibilities. Homer, Kurtz, and Royer [5] have proven that a language is $\leq^{\mathrm{P}}_{1-tt}$-hard for E if and only if it is $\leq^{\mathrm{P}}_{m}$-hard for E. It follows immediately that the language C given by the Main Theorem is weakly $\leq^{\mathrm{P}}_{1-tt}$-complete, but not $\leq^{\mathrm{P}}_{1-tt}$-complete, for E. That is, the Main Theorem holds with $\leq^{\mathrm{P}}_{m}$ replaced by $\leq^{\mathrm{P}}_{1-tt}$. Beyond this, little is known. New techniques may be required to determine whether the Main Theorem holds with $\leq^{\mathrm{P}}_{m}$ replaced by $\leq^{\mathrm{P}}_{\mathrm{T}}$.

Acknowledgments. I thank Jim Royer and Tom Linton for several useful suggestions on an earlier draft of this paper. I also thank two anonymous referees for several corrections and helpful suggestions.

References

[1] L. Berman and J. Hartmanis, On isomorphism and density of NP and

other complete sets, *SIAM Journal on Computing* **6** (1977), pp. 305–322.

[2] P. R. Halmos, *Measure Theory*, Springer-Verlag, 1950.

[3] J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms, *Transactions of the American Mathematical Society* **117** (1965), pp. 285–306.

[4] F. C. Hennie and R. E. Stearns, Two-tape simulation of multitape Turing machines, *Journal of the ACM* **13** (1966), pp. 533–546.

[5] S. Homer, S. Kurtz, and J. Royer, On 1-truth-table-hard languages, *Theoretical Computer Science* **115** (1993), pp. 383–389.

[6] D. T. Huynh, Some observations about the randomness of hard problems, *SIAM Journal on Computing* **15** (1986), pp. 1101–1105.

[7] D. W. Juedes and J. H. Lutz, The complexity and distribution of hard problems, *SIAM Journal on Computing*, to appear. See also *Proceedings of the* 34th *IEEE Symposium on Foundations of Computer Science* , Palo Alto, CA, 1993, pp. 177–185. IEEE Computer Society Press.

[8] J. H. Lutz, Resource-bounded measure, in preparation.

[9] J. H. Lutz, Category and measure in complexity classes, *SIAM Journal on Computing* **19** (1990), pp. 1100–1131.

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

[11] J. H. Lutz and E. Mayordomo, Cook versus Karp-Levin: Separating completeness notions if NP is not small, *Theoretical Computer Science*, to appear. See also *Proceedings of the Eleventh Symposium on Theoretical Aspects of Computer Science*, Springer–Verlag, 1994, pp. 415–426.

[12] J. H. Lutz and E. Mayordomo, Measure, stochasticity, and the density of hard languages, *SIAM Journal on Computing* **23** (1994), pp. 762–779.

[13] N. Lynch, On reducibility to complex or sparse sets, *Journal of the ACM* **22** (1975), pp. 341–345.

[14] E. Mayordomo, Almost every set in exponential time is P-bi-immune, *Theoretical Computer Science*, to appear. Also in *Seventeenth International Symposium on Mathematical Foundations of Computer Science*, 1992, pp. 392–400, Springer-Verlag.

[15] A. R. Meyer, 1977, reported in [1].

[16] P. Orponen and U. Schöning, The density and complexity of polynomial cores for intractable sets, *Information and Control* **70** (1986), pp. 54–68.

[17] J. C. Oxtoby, *Measure and Category*, Springer-Verlag, 1980, second edition.

[18] H. L. Royden, *Real Analysis*, Macmillan, 1968, second edition.

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

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

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

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

[23] U. Schöning, Complete sets and closeness to complexity classes, *Mathematical Systems Theory* **19** (1986), pp. 29–41.

[24] L. Stockmeyer and A. K. Chandra, Provably difficult combinatorial games, *SIAM Journal on Computing* **8** (1979), pp. 151–174.

[25] L. J. Stockmeyer, Classifying the computational complexity of problems, *Journal of Symbolic Logic* **52** (1987), pp. 1–43.