

SUBCOMPUTABLE SCHNORR RANDOMNESS

CLAUDE SURESON

IMJ, CNRS & Université Paris 7 Denis Diderot, France.
e-mail address: sureson@math.univ-paris-diderot.fr

ABSTRACT. The notion of Schnorr randomness refers to computable reals or computable functions. We propose a version of Schnorr randomness for subcomputable classes and characterize it in different ways: by Martin-Löf tests, martingales or measure computable machines.

1. INTRODUCTION

Martin-Löf randomness [Mar66] is the standard notion of randomness for infinite binary sequences. Its original definition appeals to measure but there exist different characterizations based on Kolmogorov-Chaitin theory of information [Cha75], [Lev74] or on the theory of martingales [Sch71a, Sch73].

A Martin-Löf test (written ML test) is a sequence $(G_n)_{n \in \mathbb{N}}$ of uniformly computably enumerable open subsets of $\{0, 1\}^{\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, the measure $\mu(G_n)$ is $\leq 2^{-n}$. An infinite binary sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ is Martin-Löf random if for every ML test $(G_n)_{n \in \mathbb{N}}$, $\xi \notin \bigcap_{n \in \mathbb{N}} G_n$ (ξ avoids all “effectively null set”).

Schnorr viewed this notion as too restrictive and proposed to consider only ML tests $(G_n)_{n \in \mathbb{N}}$ such that the sequence $(\mu(G_n))_{n \in \mathbb{N}}$ is uniformly computable. He also obtained a characterization in terms of martingales and orders. More recently Downey and Griffiths [DG04] characterized Schnorr’s notion using Kolmogorov complexity for “computable measure machines” (a prefix-free machine is measure computable if the measure Ω_M of the open set generated by the domain of M is computable).

One thus has (by Schnorr, Downey, Griffiths): for any $\xi \in \{0, 1\}^{\mathbb{N}}$,

ξ is Schnorr random iff for any ML test $(G_n)_{n \in \mathbb{N}}$ with $(\mu(G_n))_{n \in \mathbb{N}}$ uniformly computable, $\xi \notin \bigcap_{n \in \mathbb{N}} G_n$.

iff for any computable martingale d and any computable order h , $d(\xi \upharpoonright i) < h(i)$ almost everywhere.

iff for any computable measure machine M , there is $b \in \mathbb{N}$ such that for any $i \in \mathbb{N}$ $K_M(\xi \upharpoonright i) > i - b$.

2010 Mathematics Subject Classification: 03D25, 68Q15.

Key words and phrases: Algorithmic Randomness, Complexity Classes of Computation.

Our work originated from the following question: “can one recast these results in the primitive recursive framework or in an even weaker one”?

Schnorr showed that one can restrict to ML tests $(G_n)_{n \in \mathbb{N}}$ such that for any $n \in \mathbb{N}$, $\mu(G_n) = 2^{-n}$, to define Schnorr’s randomness. Similarly Downey and Griffiths proved that one can restrict to machines M such that $\Omega_M = 1$. Hence the natural amendments do not work (requiring $(\mu(G_n))_{n \in \mathbb{N}}$ to be “uniformly primitive recursive” or Ω_M to be a “primitive recursive real”). But for a subcomputable class \mathbf{C} of functions, one can nevertheless define a notion of ML- \mathbf{C} -S(chnorr) test and a notion of measure \mathbf{C} computable machine (one focuses on the pace of obtention of the measure).

If in the martingale formulation, we allow all martingales and orders in \mathbf{C} , the notion will be too strong and we shall not obtain equivalence with the other characterizations. Let h be a computable order. Then its inverse Inv_h (defined as $\text{Inv}_h(n) = \text{least } k \text{ } h(k) \geq n$) is also computable. It is not true anymore for primitive recursive functions. Hence, relatively to a class \mathbf{C} of functions, we shall call an order h a true \mathbf{C} -order if both h and Inv_h belong to \mathbf{C} .

In this article, we shall study the relations between the three following notions: for $\xi \in \{0, 1\}^{\mathbb{N}}$,

- ξ is ML- \mathbf{C} -S random iff ξ passes all ML- \mathbf{C} -S tests.
- ξ is Kolmogorov- \mathbf{C} -S random iff for any measure \mathbf{C} computable machine M , there is $b \in \mathbb{N}$ such that for any $n \in \mathbb{N}$, $K_M(\xi \upharpoonright n) > n - b$.
- ξ is martingale- \mathbf{C} -S random iff $\left(\begin{array}{l} \text{for any martingale } d : \{0, 1\}^* \rightarrow \mathbb{Q}_2 \text{ in } \mathbf{C}, \\ \text{and any true } \mathbf{C}\text{-order } h, d(\xi \upharpoonright i) < 2^{h(i)} \text{ a.e.} \end{array} \right.$

We show that if \mathbf{C} is the class of primitive recursive functions or the class **PSPACE**, then these three notions coincide.

One can check by using ML- \mathbf{C} -S tests that (ML) **PRIM-REC**-S randomness is strictly weaker than Schnorr randomness. But the martingale approach is better suited to separate the different notions of randomness as \mathbf{C} varies among time-complexity classes. We shall thus rely on the important amount of work centered around the martingale tool and the associated notions of subcomputable randomness. This is the field of Resource Bounded Randomness initiated by Lutz and developed by Ambos-Spies, Lutz, Mayordomo, Wang and other people.

We shall compare our notion of martingale- \mathbf{C} -S randomness with Lutz [Lut92, Lut90] notion of p-randomness, with Wang’s (**P,P**)-S randomness and with Buss, Cenzer and Remmel [BCR14] weaker notion of BP-randomness ([BCR14] results about primitive recursiveness have been a strong motivation to us). Wang’s notion is a version of Schnorr randomness for the class of polynomial time computable functions, the martingales are required to be in **P** and all orders in **P** are allowed. (By a delaying computation argument) this is the same as allowing all computable orders. Our concern with the status of the inverse of the order weakens the notion and enables more variety inside the set of computable sequences. Building on techniques of Wang and results of Schnorr, we show that one can obtain a whole hierarchy.

The two following tableaux summarize the situation: \mathbf{C} randomness is the analog for the class \mathbf{C} of computable randomness, martingale- \mathbf{C} -S randomness is abbreviated to \mathbf{C} -S randomness, and \mathbf{C} -W randomness stands for weak (Kurz) randomness with regard to the class \mathbf{C} . Implications in the tableaux cannot be reversed and in the second tableau, this

holds even when restricting to the class of computable infinite sequences.

Computable randomness	\Rightarrow	Schnorr randomness	\Rightarrow	weak randomness
\Downarrow		\Downarrow		\Downarrow
PRIM-REC randomness	\Rightarrow	PRIM-REC -S randomness	\Rightarrow	PRIM-REC -W randomness

Tableau 1

PRIM-REC randomness			\Rightarrow	PRIM-REC -S randomness			\Rightarrow	PRIM-REC -W randomness		
\Downarrow				\Downarrow				\Downarrow		
EXP randomness			\Rightarrow	EXP -S randomness			\Rightarrow	EXP -W randomness		
\Downarrow				\Downarrow				\Downarrow		
P randomness			\Rightarrow	P -S randomness			\Rightarrow	P -W randomness		

Tableau 2

2. A FEW CLASSICAL DEFINITIONS.

2.1. Some notation. \mathbb{N} , \mathbb{Q} , \mathbb{Q}_2 , \mathbb{R} denote respectively the set of natural, rational, dyadic rational and real numbers (dyadic rational numbers are of the form $m2^{-n}$, for $m \in \mathbb{Z}$ and $n \in \mathbb{N}$).

$\{0, 1\}^*$ is the set of finite binary sequences (or strings on $\{0, 1\}$) and $\{0, 1\}^{\mathbb{N}}$ is the set of infinite binary sequences.

- If x is a finite sequence, then $|x|$ represents its length. For an integer $i \in \mathbb{N}$, $x \upharpoonright i$ is the restriction of x onto the set $\{0, 1, \dots, i-1\}$.

We consider the (prefix) partial ordering \preceq defined on finite binary sequences by

$$x \preceq y \quad \text{iff} \quad x \text{ is a prefix of } y \quad (\text{that is iff } |x| \leq |y| \text{ and } y \upharpoonright |x| = x).$$

We shall also use the well-ordering \preceq_{lex} (length-lexicographic ordering):

$$x \preceq_{\text{lex}} y \quad \text{iff} \quad \begin{cases} |x| < |y| \text{ or} \\ |x| = |y| \text{ and } x \text{ is before } y \text{ in lexicographic order.} \end{cases}$$

\prec (respectively \prec_{lex}) denotes the corresponding strict ordering.

- Now for $\alpha \in \{0, 1\}^{\mathbb{N}}$ and $i \in \mathbb{N}$, we also write $\alpha \upharpoonright i$ for the restriction of α onto the set $\{0, 1, \dots, i-1\}$. If $x \in \{0, 1\}^*$, the notation $x \preceq \alpha$ means $\alpha \upharpoonright |x| = x$.

If $x, y \in \{0, 1\}^*$, $i \in \{0, 1\}$, $\alpha \in \{0, 1\}^{\mathbb{N}}$, we write xy , xi , $x\alpha$ for the corresponding concatenation.

- Given a finite set X , $|X|$ is the number of elements of X . To avoid confusion, for $r \in \mathbb{R}$, we shall write $\|r\|$ to mean the absolute value of r .

The function $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is the classical polynomial time bijection defined as

$$\langle m, n \rangle = m + (m + n)(m + n + 1)/2, \text{ for } m, n \in \mathbb{N}.$$

Let $(\cdot)_0, (\cdot)_1 : \mathbb{N} \rightarrow \mathbb{N}$ denote the (polynomial time) inverse functions: for $i \in \mathbb{N}$, $\langle (i)_0, (i)_1 \rangle = i$.

- Our references in Recursion Theory are [Odi92, Odi99], and in Algorithmic Randomness, we rely on [DH10] and [Nie09]. We thus write K_M for Kolmogorov complexity when considering a prefix-free Turing machine M (see [DH10, Ch.3.5]).

The terms “recursive” and “computable” have similar meanings.

- Concerning topology and measure, we consider the classical product topology on $\{0, 1\}^{\mathbb{N}}$ (see [DH10, Nie09]). If $x \in \{0, 1\}^*$, then we denote by $[x]$ the basic open set $\{x\alpha : \alpha \in \{0, 1\}^{\mathbb{N}}\}$ and if $X \subseteq \{0, 1\}^*$, $[X]$ is the open subset of $\{0, 1\}^{\mathbb{N}}$ generated by X , that is

$$[X] = \{x\alpha : x \in X, \alpha \in \{0, 1\}^{\mathbb{N}}\}.$$

μ is the uniform measure on $\{0, 1\}^{\mathbb{N}}$: if $x \in \{0, 1\}^*$, then $\mu([x]) = 2^{-|x|}$. When computing measure, we shall always deal with open (and hence measurable) sets.

- Generally, we use lowercase greek letters α, ξ, \dots for infinite binary sequences and lowercase roman letters x, y, \dots for finite sequences.

2.2. Schnorr Randomness. We recall here the definition of Schnorr randomness and give three different characterizations (due to Schnorr, Downey and Griffiths). For the definitions of a “computable real” or of a “computable (real valued) function”, we refer to [DH10, 5.1 and 5.2.1].

Definition 2.1.

- A sequence $(G_n)_{n \in \mathbb{N}}$ of open subsets of $\{0, 1\}^{\mathbb{N}}$ is a Martin-Löf test (abbreviated as ML test) if there is a recursively enumerable set $X \subseteq \mathbb{N} \times \{0, 1\}^*$ such that setting, for $n \in \mathbb{N}$, $X_n = \{x \in \{0, 1\}^* : (n, x) \in X\}$, one has $G_n = [X_n]$ and $\mu(G_n) \leq 2^{-n}$.
- A sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ passes the ML test $(G_n)_{n \in \mathbb{N}}$ if $\xi \notin \bigcap_{n \in \mathbb{N}} G_n$ (otherwise it fails the test).
- A sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ is random if it passes all ML tests.

Schnorr viewed this notion of randomness as too strong and proposed the following:

Definition 2.2 (Schnorr).

- A Schnorr test is an ML test $(G_n)_{n \in \mathbb{N}}$ such that $\mu(G_n)$ is uniformly computable in n .
- A sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ is Schnorr random if it passes all Schnorr tests.

There is a characterization of Schnorr randomness in terms of martingales. We recall:

Definition 2.3.

- A function $d : \{0, 1\}^* \rightarrow \mathbb{R}^+$ is a martingale if for any $x \in \{0, 1\}^*$, $d(x0) + d(x1) = 2d(x)$.
- A function $h : \mathbb{N} \rightarrow \mathbb{N}$ is an order if it is nondecreasing and unbounded.

Theorem 2.4 ([Sch71b]). *A sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ is Schnorr random iff for any computable martingale d and any computable order h , $d(\xi \upharpoonright n) < h(n)$ a.e. (a.e. stands for “almost everywhere”).*

The last characterization we shall consider in this article is more recent and due to Downey and Griffiths.

Definition 2.5.

- If M is a Turing machine, then Ω_M is the measure $\mu([\text{dom}(M)])$.
- A prefix-free machine M is called a computable measure machine if Ω_M is a computable real.

Theorem 2.6 ([DG04]). *A sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ is Schnorr random iff for each computable measure machine M , there is $b \in \mathbb{N}$ such that for any n , $K_M(\xi \upharpoonright n) > n - b$.*

When trying to extend these definitions to subcomputable classes, one must be cautious. For instance, the notion of “primitive recursive real” is problematic (see [CSZ07]). We shall consider Cauchy style definitions (rather than left cut ones) and all the definitions which have been omitted in this review paragraph, will be provided.

3. NOTIONS OF RESOURCE BOUNDED SCHNORR RANDOMNESS.

An important body of concepts and results (now classical) has been obtained by Ambos-Spies, Ko, Lutz, Mayodormo, Wang and others (see [AM97] for a survey). Our original motivation came from primitive recursiveness and the article of Buss, Cenzer and Remmel [BCR14]. We shall thus build on all these works to propose here, relatively to a class of functions, three possible characterizations (inspired from the previous section) of Schnorr resource bounded randomness: in terms of Martin-Löf tests, Kolmogorov complexity and martingales. (Depending on the chosen class) we shall study when these different approaches lead to the same notion. Later in the paper, we shall also compare these definitions with a different concept proposed by Wang [Wan00].

To motivate our definitions, let us note that some properties are given for free when dealing with recursive functions. For instance, if $f : \mathbb{N} \rightarrow \mathbb{N}$ is recursive and unbounded, its inverse Inv_f (defined by $Inv_f(n) = \text{least } k \text{ } f(k) \geq n$) is also recursive. It is not true anymore for primitive recursive functions: there exists a primitive recursive function whose inverse is the Ackermann function $n \mapsto A(n, n)$. Hence our definitions will have to incorporate new conditions.

3.1. Definitions. We shall consider time-complexity classes \mathbf{C} of functions of the form:

$$\mathbf{C} = \bigcup_{f \in F_C} \text{FDTIME}(f(n)),$$

for F_C a class with appropriate closure properties of time-constructible functions.

Such classes \mathbf{C} are:

Definition 3.1.

- (1) $\mathbf{P} = \bigcup_{k \in \mathbb{N}} \text{FDTIME}(n^k)$,
- (2) $\mathbf{EXP} = \bigcup_{k \in \mathbb{N}} \text{FDTIME}(2^{n^k})$,
- (3) Let $T : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ be the function recursively defined by

$$\begin{aligned} T(0, n) &= n \\ T(k+1, n) &= 2^{T(k, n)}. \end{aligned}$$

Then $\mathbf{TOWER-EXP} = \bigcup_{k \in \mathbb{N}} \text{FDTIME}(T(k, n))$.

- (4) $\mathbf{PRIM-REC}$ is the class of recursively primitive functions.

To view $\mathbf{PRIM-REC}$ as a time-complexity class may require some justification: an easy modification of [Odi99, Thm VIII 8.8]) gives:

Lemma 3.2. *A (total) function f is primitive recursive iff it can be computed by a Turing machine in time $\mathcal{O}(g(n))$, for g primitive recursive and time-constructible.*

For example, if $\mathbf{C} = \mathbf{P}$, then we take for F_C the class of polynomial functions with coefficients in \mathbb{N} ; if $\mathbf{C} = \mathbf{PRIM-REC}$, then we consider for F_C the class of primitive recursive functions which are time-constructible.

Definition 3.3. If $\mathbf{C} = \bigcup_{f \in F_C} \text{FDTIME}(f(n))$, with F_C as above, then let $\mathbf{EXP}(\mathbf{C}) = \bigcup_{f \in F_C} \text{FDTIME}(2^{f(n)})$.

For instance, $\mathbf{EXP} = \mathbf{EXP}(\mathbf{P})$ and $\mathbf{C} = \mathbf{EXP}(\mathbf{C})$ when \mathbf{C} is **TOWER-EXP** or **PRIM-REC** (all this will allow us to state our results in a unified way). We shall also consider the classical space related class:

Definition 3.4. $\mathbf{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{FSPACE}(n^k)$.

When dealing with functions $f : \mathbb{N} \rightarrow \mathbb{N}$, martingales $d : \{0, 1\}^* \rightarrow \mathbb{Q}_2$ or approximations of \mathbb{R} valued martingales $g : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}_2$, to decide whether these functions belong to one of the above classes \mathbf{C} , we must fix a representation of the different inputs and outputs, and hence a measurement of their size.

- integers will be under unary representation: $n \in \mathbb{N}$ is thus viewed as 1^n and its size is n ,
- strings $x \in \{0, 1\}^*$ have classically size $|x|$,
- there is a constant $\theta \in \mathbb{N}$ such that dyadic rational numbers of the form $m2^{-n}$, for $m \in \mathbb{N}$ and $n \in \mathbb{N}$, are (reasonably) coded by a string in $\{0, 1\}^*$ of length $\leq \theta(\log(m) + n)$

To introduce subcomputable Martin-Löf tests, let us state a few definitions. We often identify a Turing machine M with the (partial) recursive function it computes. We write $M(x) \downarrow$ to mean that the machine M halts on input x (yielding as output whatever is written on a dedicated tape).

Definition 3.5.

- (a) Let M be a Turing machine. Then for $t \in \mathbb{N}$, $x, y \in \{0, 1\}^*$, we set
- $M_t(x) = y$ iff ($M(x) \downarrow$ and on input x , M outputs y in at most t steps.
 - $M_t^{\text{space}}(x) = y$ iff ($M(x) \downarrow$ and on input x , M outputs y having used at most t cells ($t \geq |x|, |y|$).
- (b) Given a recursively enumerable set $X \subseteq \mathbb{N} \times \{0, 1\}^*$ and a machine M such that $X = \text{dom}(M)$, we set for $m, t \in \mathbb{N}$,
- $X_m = \{x \in \{0, 1\}^* : (m, x) \in X\} = \{x \in \{0, 1\}^* : M(m, x) \downarrow\}$
 - $X_{m,t}^M = \{x \in \{0, 1\}^* : (m, x) \in \text{dom}(M_t)\}$,
 - $X_{m,t}^{M,\text{space}} = \{x \in \{0, 1\}^* : (m, x) \in \text{dom}(M_t^{\text{space}})\}$.

Let $(G_n)_{n \in \mathbb{N}}$ be a Schnorr test (Definition 2.2): the sequence $(\mu(G_n))_{n \in \mathbb{N}}$ is thus uniformly computable. This implies the existence of a computable function $F : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{Q}_2$ such that for any $i, n \in \mathbb{N}$, $\|\mu(G_n) - F(n, i)\| \leq 2^{-i}$.

Hence a natural attempt to extend the notion of ML Schnorr test to the primitive recursive context would be to require F to be primitive recursive and to consider as random, infinite sequences which pass all such tests . This is doomed:

Theorem 3.6 ([Sch71b]). *Let $(G_n)_{n \in \mathbb{N}}$ be a Schnorr test. Then there exists a Schnorr test $(O_n)_{n \in \mathbb{N}}$ such that for any n , $\mu(O_n) = 2^{-n}$ and $\bigcap_{n \in \mathbb{N}} G_n \subseteq \bigcap_{n \in \mathbb{N}} O_n$.*

Hence ML tests $(G_n)_{n \in \mathbb{N}}$ with $\mu(O_n) = 2^{-n}$, for $n \in \mathbb{N}$, suffice to define Schnorr randomness.

Another try consists in noting (it is implicit in several classical proofs) that if $(G_n)_{n \in \mathbb{N}}$ is a Schnorr test - associated with $X \subseteq \mathbb{N} \times \{0, 1\}^*$ and a machine M - then using the approximating function F above, one can check the existence of a computable function $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that

- for any $n, i \in \mathbb{N}$, $\mu([X_n]) - \mu([X_{n,f(n,i)}^M]) \leq 2^{-i}$.

We could thus require f to be primitive recursive and consider the following definition:

Definition 3.7.

- Let \mathbf{C} be one of our time-complexity classes. An ML test $(G_n)_{n \in \mathbb{N}}$ is called an ML- \mathbf{C} -S test if there exist a recursively enumerable set $X \subseteq \mathbb{N} \times \{0, 1\}^*$ associated with a machine M such that $X = \text{dom}(M)$ and a function $f : \mathbb{N} \rightarrow \mathbb{N}$ in \mathbf{C} , called the controlling function, such that for any $m, i \in \mathbb{N}$,
 - $G_m = [X_m]$,
 - $\mu([X_m]) - \mu([X_{m,f(m+i)}^M]) \leq 2^{-i}$
- For $\mathbf{C} = \mathbf{PSPACE}$, we require $f : \mathbb{N} \rightarrow \mathbb{N}$ to be in \mathbf{PSPACE} and to satisfy for any $m, i \in \mathbb{N}$, $\mu([X_m]) - \mu([X_{m,f(m+i)}^{M,\text{space}}]) \leq 2^{-i}$

(The “S” in ML- \mathbf{C} -S-test stands for “Schnorr”)

Concerning an extension of the Downey-Griffiths characterization, we could restrict to prefix-free machines M such that Ω_M is a primitive recursive real. But again, this does not produce a new notion since by [DG04], to characterize Schnorr randomness, one can restrict to machines N with $\Omega_N = 1$. Hence we proceed as in the previous definition.

Definition 3.8. Given a prefix-free machine M and $t \in \mathbb{N}$, we set

$$\Omega_{M_t} = \mu([\text{dom}(M_t)]) \quad \text{and} \quad \Omega_{M_t^{\text{space}}} = \mu([\text{dom}(M_t^{\text{space}})]).$$

- For \mathbf{C} one of our time-complexity classes, a prefix-free machine M is termed “measure \mathbf{C} computable” if there is a function $g : \mathbb{N} \rightarrow \mathbb{N}$ in \mathbf{C} , (also) called the controlling function, such that for any $i \in \mathbb{N}$, $\Omega_M - \Omega_{M_{g(i)}} \leq 2^{-i}$.
- If $\mathbf{C} = \mathbf{PSPACE}$, we require the existence of g in \mathbf{PSPACE} such that $\Omega_M - \Omega_{M_{g(i)}^{\text{space}}} \leq 2^{-i}$.

Remark 3.9.

- Given \mathbf{C} a time complexity-class, since integers are under unary representation, one gets the same notion of test or of measure \mathbf{C} -computability by requiring the controlling function to be in \mathbf{C} or in $F_{\mathbf{C}}$.
- Similarly for \mathbf{PSPACE} , one can indifferently require the controlling function to be a polynomial function (with coefficients in \mathbb{N}), a function in \mathbf{P} or a function in \mathbf{PSPACE} .

As mentioned earlier, part(ii) of the following definition is redundant in the recursive case:

Definition 3.10.

- Given an unbounded function $f : \mathbb{N} \rightarrow \mathbb{N}$, one defines the “inverse” of f as follows: for $n \in \mathbb{N}$, $\text{Inv}_f(n) = \text{least } k \text{ s.t. } f(k) \geq n$.
- Let \mathbf{C} be one of our complexity classes, an order h is a true \mathbf{C} -order if both h and Inv_h belong to \mathbf{C} .

In the subcomputable framework, to define martingale related randomness, we shall restrict to \mathbb{Q}_2 -valued martingales (this is not absolutely necessary, one can consider \mathbb{R} -valued martingales which are \mathbf{C} -approximable, as does Lutz [Lut92] for $\mathbf{C} = \mathbf{P}$).

We now state the respective definitions of \mathbf{C} -Schnorr randomness.

Definition 3.11. Let \mathbf{C} be one of our complexity classes and let $\xi \in \{0, 1\}^{\mathbb{N}}$.

- (a) ξ is ML- \mathbf{C} -S random iff ξ passes all ML- \mathbf{C} -S tests.
- (b) ξ is Kolmogorov- \mathbf{C} -S random iff for any measure \mathbf{C} computable machine M , there is $b \in \mathbb{N}$ such that for any $n \in \mathbb{N}$, $K_M(\xi \upharpoonright n) > n - b$.
- (c) ξ is martingale- \mathbf{C} -S random iff $\left(\begin{array}{l} \text{for any martingale } d : \{0, 1\}^* \rightarrow \mathbb{Q}_2 \text{ in } \mathbf{C}, \\ \text{and any true } \mathbf{C}\text{-order } h, d(\xi \upharpoonright i) < 2^{h(i)} \text{ a.e.} \end{array} \right.$

Remark 3.12. If \mathbf{C} is **TOWER-EXP** or **PRIM-REC**, then the condition “ $d(\xi \upharpoonright i) < 2^{h(i)}$ a.e.” gives the same notion of randomness as the usual one “ $d(\xi \upharpoonright i) < h(i)$ a.e.”.

We shall first study the relation between Martin-Löf and Kolmogorov complexity notions of randomness, and later the link between Martin-Löf and martingale notions of randomness.

3.2. The relation between the Martin-Löf and the Kolmogorov complexity notions. In our definition of ML- \mathbf{C} -S tests, we did not require the generating sets to be prefix-free. To obtain this in a uniform way, we shall resort to the classical argument showing that a recursively enumerable generating set can be replaced by a recursive prefix-free one (see [Nie09, 1.8.26]). We propose here a quadratic time algorithm (or linear space) algorithm yielding the new generating set.

Claim 3.13. Let $X \subseteq \mathbb{N} \times \{0, 1\}^*$ and let M be a machine such that $\text{dom}(M) = X$. Then one can define a set $Y \subseteq \mathbb{N} \times \{0, 1\}^*$ and a machine N such that:

- (a) $Y = \{(n, x) \in \mathbb{N} \times \{0, 1\}^* : N(n, x) \downarrow\}$ and there is a constant $d \in \mathbb{N}$ such that for any (n, x) , $N(n, x) \downarrow \Leftrightarrow N$ on input (n, x) halts in at most $d(n + |x|)^2$ steps.
- (b) Setting for $n, t \in \mathbb{N}$, $\left(\begin{array}{l} Y_n = \{x \in \{0, 1\}^* : N(n, x) \downarrow\} \text{ and} \\ Y_n(t) = \{x \in \{0, 1\}^* : |x| = t \text{ and } x \in Y_n\}, \end{array} \right.$ one has $[X_{n,t}^M] = [Y_n(t)]$.
- (c) $Y_n = \bigcup_{s \in \mathbb{N}} Y_n(s)$ is prefix free.
- (d) For any $n, s \in \mathbb{N}$, $[X_n] = [Y_n]$ and $[X_{n,s}^M] \subseteq [Y_{n,d(n+s)^2}^N]$.

Proof. (Sketch)

- (a) Let us consider the following algorithm for the machine N : on input (n, x)
 - (1) if there is $y \preceq x$ such that $y \in X_{n,|x|-1}^M$, then N rejects the input (may loop indefinitely),
 - (2) otherwise (2.1) if there is $y \preceq x$ such that $y \in X_{n,|x|}^M$, then N halts.
 - (2.2) otherwise, N rejects (n, x) .

If N halts on an input, it does so in quadratic time. Let us set $Y = \text{dom}(N)$.

- (b) One can check by induction on $s \in \mathbb{N}$, $[X_{n,s}^M] \subseteq [Y_{n,d(n+s)^2}^N]$.

The rest follows. □

Concerning the class **PSPACE**, we note that if we replace $X_{n,t}^M$ in the above algorithm by $X_{n,t}^{M,\text{space}}$, the algorithm requires linear space. We thus get:

Claim 3.14. Under the same hypotheses as in the previous claim, we obtain $Y \subseteq \mathbb{N} \times \{0, 1\}^*$ and a constant $d \in \mathbb{N}$ such that for any $n, s \in \mathbb{N}$ and

- $[X_n] = [Y_n]$, Y_n prefix-free and
- $[X_{n,s}^{M,\text{space}}] \subseteq [Y_{n,d(n+s)^2}^{N,\text{space}}]$.

Remark 3.15.

- (1) As a consequence of these two claims, when considering our complexity class \mathbf{C} , in the definition of ML- \mathbf{C} -S tests $(G_n)_{n \in \mathbb{N}}$ (def 3.7), we shall assume the generating sets $X_n \subseteq \{0, 1\}^*$, for $n \in \mathbb{N}$, to be prefix-free.
- (2) Let us also note that if the controlling function $f : \mathbb{N} \rightarrow \mathbb{N}$ is in \mathbf{C} , then the function $g : \mathbb{N} \rightarrow \mathbb{N}$ defined recursively by
$$\begin{cases} - g(0) = f(0), \\ - g(n+1) = \max\{g(n) + 1, f(n+1)\} \end{cases}$$
 is also in \mathbf{C} (for any $n \in \mathbb{N}$, $g(n) \leq \max\{f(m) : m \leq n\} + n$) and since $f \leq g$, it also satisfies for any $n, i \in \mathbb{N}$, $\mu([X_n]) - \mu([X_{n, g(n+i)}^M]) \leq 2^{-i}$. Hence we shall assume the controlling function to be strictly increasing.
- (3) This also applies to the controlling function in the definition of the measure \mathbf{C} computable machine.

Following the notation in [Nie09, 3.2.6], we recall:

Definition 3.16. Given a prefix-free Turing machine M and $b \in \mathbb{N}$, one considers the subset R_b^M of $\{0, 1\}^{\mathbb{N}}$ defined as:

$$R_b^M = [\{x \in \{0, 1\}^* : K_M(x) \leq |x| - b\}] = \{\alpha \in \{0, 1\}^{\mathbb{N}} : \exists k K_M(\alpha \upharpoonright k) \leq k - b\}.$$

We shall show:

Proposition 3.17.

- (a) Let \mathbf{C} be one of our time-complexity classes and let $(G_m)_{m \in \mathbb{N}}$ be an ML- \mathbf{C} -S test. Then there exists a measure **EXP**(\mathbf{C})-computable machine M such that

$$\bigcap_{m \in \mathbb{N}} G_m \subseteq \bigcap_{b \in \mathbb{N}} R_b^M$$

- (b) If $(G_m)_{m \in \mathbb{N}}$ is an ML-**PSPACE**-S test, then there is a measure **PSPACE** computable machine M such that the above inclusion holds.

We shall push classical arguments as far as possible (see [Nie09, 3.5.18]). But a blind application of the Kraft-Chaitin theorem will not suffice to produce the adequate measure machine M (we would only get Ω_M rightly approximable and we need more control than that). Hence after this exploration, we shall define a new goal and a strategy to reach it.

Proof.

- (a) Let \mathbf{C} be one of our time-complexity classes and let $(G_n)_{n \in \mathbb{N}}$ be an ML- \mathbf{C} -S test. (By replacing $(G_n)_{n \in \mathbb{N}}$ by $(G_{2n})_{n \in \mathbb{N}}$, the controlling function f by $n \mapsto f(2n)$, and using the inclusion $\bigcap_{n \in \mathbb{N}} G_n \subseteq \bigcap_{n \in \mathbb{N}} G_{2n}$) we can assume there exist $X \subseteq \mathbb{N} \times \{0, 1\}^*$ associated with a machine N , and a strictly increasing function f in \mathbf{C} such that for any $n, i \in \mathbb{N}$,
- (i) $X_n = \{x \in \{0, 1\}^* : (n, x) \in X\} = \{x \in \{0, 1\}^* : N(n, x) \downarrow\}$ is prefix-free
 - (ii) $G_n = [X_n]$ and $\mu(G_n) \leq 2^{-2n}$
 - (iii) $\mu([X_n]) - \mu([X_{n, f(n+i)}^N]) \leq 2^{-i}$.

Let us consider the bounded request set

$$L = \{(|x| - m + 1, x) : x \in X_m\}. \quad (3.1)$$

Its weight $\gamma_L = \sum_{(m, x) \in X} 2^{-(|x| - m + 1)}$ is ≤ 1 by (ii).

Let us apply the Kraft-Chaitin theorem (see [Nie09, 2.2.17], [DH10, 3.6.1]) to L (starting with an effective enumeration of X and hence of L) to see how far it can bring us:

there is machine M such that the following properties hold:

$$\left\{ \begin{array}{l} - M \text{ is prefix-free,} \\ - \Omega_M = \gamma_L, \\ - \text{for every } (m, x) \in X, \text{ there is a unique } w(m, x) \in \{0, 1\}^* \text{ such that} \\ \quad M(w(m, x)) = x \text{ and } |w(m, x)| = |x| - m + 1, \\ - \text{dom}(M) = \{w(m, x) : x \in X_m\}. \end{array} \right. \quad (3.2)$$

Let us check that Ω_M can be approximated by a sequence $s : \mathbb{N} \rightarrow \mathbb{Q}_2$ in **EXP(C)**, that is, for any $r \in \mathbb{N}$, one has $\|\Omega_M - s(r)\| \leq 2^{-r}$. This will not suffice, but it may give us some insight on how to replace $s(r)$ by $\Omega_{M_{h(r)}}$, for h in **EXP(C)**.

One has $\gamma_L = \Omega_M = \sum_{(m,x) \in X} 2^{-(|x|-m+1)} = \sum_m \sum_{x \in X_m} 2^{-(|x|-m+1)}$.

For $r \in \mathbb{N}$, let us set $\gamma_r = \sum_{m \leq r} \sum_{x \in X_{m,f(3r+1)}^N} 2^{-(|x|-m+1)}$. Let r be fixed.

- If $m \leq r$, then $f(m+2r+1) \leq f(3r+1)$. Hence for $m \leq r$

$$\mu([X_m \setminus X_{m,f(3r+1)}^N]) = \mu([X_m]) - \mu([X_{m,f(3r+1)}^N]) \leq 2^{-(2r+1)}.$$

The first equality holds because X_m is prefix-free.

- We deduce:

$$\begin{aligned} \gamma_L - \gamma_r &= \sum_m \sum_{x \in X_m} 2^{-(|x|-m+1)} - \sum_{m \leq r} \sum_{x \in X_{m,f(3r+1)}^N} 2^{-(|x|-m+1)} \\ &\leq \sum_{m \leq r} \sum_{x \in X_m \setminus X_{m,f(3r+1)}^N} 2^{-(|x|-m+1)} + \sum_{m > r} \sum_{x \in X_m} 2^{-(|x|-m+1)} \\ &\leq \sum_{m \leq r} 2^{m-1} 2^{-(2r+1)} + \sum_{m > r} 2^{m-1} \mu(G_m) \\ &\leq 2^{-r}. \end{aligned} \quad (3.3)$$

We note that the function $\varphi : \mathbb{N} \rightarrow \mathbb{Q}_2$ belongs to **EXP(C)** (for all z such that $r \mapsto \gamma_r$).

$|z| \leq f(3r+1)$, we have to check whether $z \in X_{m,f(3r+1)}^N$.

Also by the properties of (3.2), for $r \in \mathbb{N}$, one has:

$$\gamma_r = \sum_{m \leq r} \sum_{x \in X_{m,f(3r+1)}^N} 2^{-|w(m,x)|}. \quad (3.4)$$

This suggests the following claim (inside the proof of Proposition 3.17, we use a local numbering with letters of definitions and claims):

Claim A. Let M be a machine satisfying the properties of (3.2) and let $h : \mathbb{N} \rightarrow \mathbb{N}$ be a function such that for any $r \in \mathbb{N}$, $\{w(m, x) : m \leq r, x \in X_{m,f(3r+1)}^N\} \subseteq \text{dom}(M_{h(r)})$. Then for any $r \in \mathbb{N}$, $\Omega_M - \Omega_{M_{h(r)}} \leq 2^{-r}$.

Proof. For $r \in \mathbb{N}$, the inclusion $\{w(m, x) : m \leq r, x \in X_{m,f(3r+1)}^N\} \subseteq \text{dom}(M_{h(r)})$ and (3.4) imply

$$\gamma_r \leq \mu([\text{dom}(M_{h(r)})]) \leq \Omega_M = \gamma_L.$$

Hence by (3.3), $\Omega_M - \Omega_{M_{h(r)}} \leq \gamma_L - \gamma_r \leq 2^{-r}$. \square

The strategy to obtain Proposition 3.17(a) (induced by the previous claim):

(a.1) We define a (c.e) well-ordering \sqsubseteq_X on X .

(a.2) We then develop a proof of Kraft-Chaitin theorem by \sqsubseteq_X -induction, in order to obtain a function $w : X \rightarrow \{0,1\}^*$ and a machine M as in (3.2) (sending $w(m, x)$ to x).

(a.3) A sufficient condition to satisfy the hypotheses of Claim A, is that M works in **EXP(C)** time (on successful computations). To define such an algorithm for M , the idea is to stratify X as $\bigcup_{r \in \mathbb{N}} X(r)$ so that $X(r)$ is a finite initial segment of X for \sqsubseteq_X and such that for any $(m, x) \in X$, if $|w(m, x)| \leq r$, then $(m, x) \in X(r)$. Hence for any $w \in \{0,1\}^*$ such that $|w| \leq r$, to check whether $w = w(m, x)$ for some $(m, x) \in X$ (and thus to send w to x), we shall only have to carry out the \sqsubseteq_X -induction process on $X(r)$.

(a.4) We finally define an algorithm for M based on the previous remark which satisfies the time bounds.

(a.1) **The definition of the well-ordering:**

Definition B.

• Let \sqsubseteq be the ordering on $\mathbb{N} \times \{0,1\}^*$ defined as follows: for $m, m' \in \mathbb{N}$, $x, x' \in \{0,1\}^*$

$$(m, x) \sqsubseteq (m', x') \quad \text{iff} \quad \begin{cases} x \prec_{\text{lex}} x' & \text{or} \\ x = x' & \text{and } m \leq m'. \end{cases}$$

• Let \sqsubseteq_X denote the restriction of \sqsubseteq on X .

Claim C. For any $(m, x) \in X$, the set $\{(m', x') \in X : (m', x') \sqsubseteq_X (m, x)\}$ is finite. Hence every element of X - except the least one - admits an immediate predecessor for \sqsubseteq_X .

Proof. Let $(m, x), (m', x') \in X$ be such that $(m', x') \sqsubseteq (m, x)$.

Necessarily $|x'| \leq |x|$. Since $x' \in X_{m'}$, we have $[x'] \subseteq [X_{m'}]$. Hence $2^{-|x'|} \leq \mu([X_{m'}]) \leq 2^{-2m'}$.

Therefore $2m' \leq |x'| \leq |x|$. □

We then set:

Definition D.

- For $(m, x) \in \mathbb{N} \times \{0,1\}^*$, let $r_{m,x} = |x| - m + 1$.
- For $(m, x) \in X$, let $\text{pred}(m, x)$ be the immediate predecessor of (m, x) for \sqsubseteq_X ; if (m_0, x_0) is the least element of X for \sqsubseteq_X , we set $\text{pred}(m_0, x_0) = (\emptyset, -1)$.
- For a set Z , let $\mathcal{P}_{\text{finite}}(Z)$ be the collection of finite subsets of Z .

(a.2) **The inductive construction** ([Nie09, 2.2.17] of **Kraft-Chaitin Theorem based on \sqsubseteq_X** :

We define by induction on \sqsubseteq_X the following functions:

$$\begin{pmatrix} R : X \rightarrow \mathcal{P}_{\text{finite}}(\{0,1\}^*) \\ w : X \rightarrow \{0,1\}^* \\ z : X \rightarrow \{0,1\}^*. \end{pmatrix}$$

– $R(\emptyset, -1) = \{\emptyset\}$.

– At step $(m, x) \in X$, we suppose $R(\text{pred}(m, x))$ is known and define $z(m, x)$, $w(m, x)$ and $R(m, x)$:

(i) Let $z(m, x)$ be the longest string in $R(\text{pred}(m, x))$ of length $\leq r_{m,x}$.

- (ii) Let $w(m, x)$ be the leftmost string (least for lexicographic order) of length $r_{m,x}$ extending $z(m, x)$ (i.e. $w(m, x) = z(m, x)0^{r_{m,x}-|z(m,x)|}$)
- (iii) One sets $R(m, x) = (R(\text{pred}(m, x)) \setminus \{z(m, x)\}) \cup \{z(m, x)0^i 1 : 0 \leq i < r_{m,x} - |z(m, x)|\}$.

By classical arguments [Nie09, 2.2.17], the construction can be carried out and is effective (by the proof of Claim C and by Claim F, there exists an effective increasing - with respect to \sqsubseteq_X - enumeration of X). Hence with $w : X \rightarrow \{0, 1\}^*$ defined as above, there is a machine M satisfying Properties (3.2) which for $(m, x) \in X$ sends $w(m, x)$ to x .

Instead of justifying the assertion about the existence of an effective increasing enumeration of X , let us define now the stratification of X which will allow us to exhibit an algorithm for M with the appropriate time bounds.

(a.3) The stratification of X :

Definition E. For $r \in \mathbb{N}$, let $X(r) = \{(m, x) : x \in X_{m,f(3r+1)}^N, |x| \leq 2r, m \leq r\}$.

The last requirement “ $m \leq r$ ” is redundant, we left it to stress the fact that $X(r)$ is finite. Let us note the following:

Claim F. If $(m, x) \in X$, then

- (a) $|x| \leq 2r_{m,x}$, $2m \leq |x|$ and $m \leq r_{m,x}$,
- (b) $x \in X_{m,f(3r_{m,x}+1)}^N$.

Proof. Let $(m, x) \in X$.

- (a) As we noted in the proof of Claim C, $x \in X_m$ implies $|x| \geq 2m$. Hence we deduce $r_{m,x} = |x| - m + 1 \geq m + 1$ and $r_{m,x} \geq |x| - (|x|/2) + 1 \geq |x|/2$.
- (b) Since $\mu([X_m]) - \mu([X_{m,f(m+|x|+1)}^N]) \leq 2^{-(|x|+1)}$, necessarily $x \in X_{m,f(m+|x|+1)}^N$.
Now by (a), $m \leq r_{m,x}$, $|x| \leq 2r_{m,x}$, hence since f increasing, $x \in X_{m,f(3r_{m,x}+1)}^N$. \square

The interest of the stratification of X appears in the following:

Claim G.

- (a) Let $r \in \mathbb{N}$. Then $(X(r), \sqsubseteq_X)$ is a finite initial segment of (X, \sqsubseteq) .
- (b) If $r \in \mathbb{N}$, $(m, x) \in X$ and $r_{m,x} \leq r$, then $(m, x) \in X(r)$.

Proof.

- (a) Let $(m', x'), (m, x)$ be both in X , $(m', x') \sqsubseteq (m, x)$ and $(m, x) \in X(r)$.
Necessarily $|x'| \leq |x|$. Hence $|x'| \leq 2r$. Also

$$(m', x') \in X \Rightarrow \begin{cases} 2m' \leq |x'| \\ x' \in X_{m',f(m'+|x'|+1)} \end{cases}$$

We deduce $m' \leq r$ and $x' \in X_{m',f(3r+1)}$. Therefore $(m', x') \in X(r)$.

- (b) Let $(m, x) \in X$ and $r_{m,x} \leq r$. Then by Claim F(b), $x \in X_{m,f(3r+1)}^N$. \square

(a.4) The algorithm for M :

The proof of the Kraft-Chaitin theorem yields the following:

Claim H. Let $r \geq 1$.

- (a) For any $(m, x) \in X(r)$, $R(m, x)$ contains at most $2r+2$ strings which are of length $\leq 2r+1$. Hence there is a constant $d \in \mathbb{N}$ such that $R(m, x)$ can be coded (according to increasing length) by a string $\rho(m, x)$ of length $\leq dr^2$.
- (b) Also if $(m', x') = \text{pred}(m, x)$ with $(m', x'), (m, x) \in X(r)$, there is an algorithm requiring $\mathcal{O}(r^2)$ steps which produces $\rho(m, x)$ from $\rho(m', x')$.

Proof. We use the classical fact that all strings have different length, adding the bound information.

- (a) Let $r \in \mathbb{N}$ be fixed. One argues by \sqsubseteq_X -induction: let $(m, x) \in X(r)$. We assume that all strings in $R(\text{pred}(m, x))$ have different length $\leq 2r+1$.
By construction (requirements (i),(iii)), we obtain:
 - $R(m, x) \setminus R(\text{pred}(m, x)) \subseteq \{z \in \{0, 1\}^* : |z(m, x)| < |z| \leq r_{m,x}\}$,
 - $R(\text{pred}(m, x)) \cap \{z \in \{0, 1\}^* : |z(m, x)| < |z| \leq r_{m,x}\} = \emptyset$,
 - All strings in $R(m, x) \setminus R(\text{pred}(m, x))$ have different length $\leq r_{m,x}$.
 Now $r_{m,x} = |x| - m + 1 \leq |x| + 1 \leq 2r + 1$.
Hence by induction hypothesis, all elements in $R(m, x)$ have different length $\leq 2r + 1$.
- (b) Also if the elements of $R(\text{pred}(m, x))$ are enumerated according to increasing length, it takes $\mathcal{O}(r^2)$ steps to build $R(m, x)$ enumerating its elements according to increasing length. \square

By Claim G, we can replace the induction on \sqsubseteq_X by an induction on $\sqsubseteq_{X(r)}$:

An algorithm for M .

- For $r \in \mathbb{N}$, let $A(r) = \{(m, x) \in \mathbb{N} \times \{0, 1\}^* : m \leq r, |x| \leq 2r\}$. We shall enumerate $A(r)$ according to \sqsubseteq . Let ρ_0 code the set $\{\emptyset\}$; ρ will be a variable whose value is $\rho(m, x)$ where (m, x) is the last element of $X(r)$ which has been treated.
- On input $w \in \{0, 1\}^*$ such that $|w| = r$,
 - set $\rho := \rho_0$,
 - $(m, x) \in A(r)$.
 - Case 1:* if $x \in X_{m, f(3r+1)}^N$, then from ρ (playing the role of $R(\text{pred}(m, x))$), compute the values $w(m, x)$ and $\rho(m, x)$, and set $\rho := \rho(m, x)$.
 - Case 1.1:* if $w(m, x) = w$, then output x and stop the machine.
 - Case 1.2:* otherwise
 - Case 1.2.1:* if $(m, x) = (r, 1^{2r-1})$ (the maximum of $A(r)$), then loop for ever,
 - Case 1.2.2:* otherwise compute the successor (m', x') of (m, x) for \sqsubseteq in $A(r)$ and go to step (m', x') .
 - Case 2:* if $x \notin X_{m, f(3r+1)}^N$,
 - Case 2.1:* if $(m, x) = (r, 1^{2r-1})$, then loop for ever,
 - Case 2.2:* otherwise, compute the successor (m', x') of (m, x) for \sqsubseteq in $A(r)$ and go to step (m', x') .

Let $\mathbf{C} = \bigcup_{g \in F_C} \text{FDTIME}(g(n))$ be one of our time-complexity classes. Since the integers are under unary representation, $f(n) = \mathcal{O}(f'(n))$ for some $f' \in F_C$. Using Claim H, one can then show the existence of a constant c_0 and of a function $g \in F_C$ so that, for any $|w| = r$, any $(m, x) \in A(r)$, if step (m, x) is finite, it takes at most $c_0 g(r)$ steps to be completed.

Now, for $r \in \mathbb{N}$, $|A(r)| \leq 2^{2r+1}(r+1)$.

The time bound: hence there exists $l \in \mathbf{C}$ such that any successful computation of M on a finite sequence w with $|w| = r$, takes at most $l(r)2^{3r}$ steps. Therefore

$$\begin{aligned} \text{dom}(M) \cap \{w \in \{0,1\}^* : |w| \leq r\} &= \{w(m,x) : (m,x) \in X \text{ and } r_{m,x} \leq r\} \\ &\subseteq \text{dom}(M_{l(r)2^{3r}}). \end{aligned} \quad (3.5)$$

Our goal is to fulfill the conditions of Claim A, that is to obtain a function h in $\mathbf{EXP}(\mathbf{C})$ so that:

$$\{w(m,x) : m \leq r, x \in X_{m,f(3r+1)}\} \subseteq \text{dom}(M_{h(r)}).$$

If $x \in X_{m,f(3r+1)}$, then $|x| \leq f(3r+1)$. Hence $r_{m,x} = |x| - m + 1 \leq f(3r+1) + 1$. Therefore

$$\begin{aligned} \{w(m,x) : m \leq r, x \in X_{m,f(3r+1)}^N\} &\subseteq \{w(m,x) : (m,x) \in X \text{ and } r_{m,x} \leq f(3r+1) + 1\} \\ &\subseteq \text{dom}(M_{l(f(3r+1)+1)2^{3(f(3r+1)+1)}}) \text{ (by (3.5)).} \end{aligned}$$

Let us define h by $h(r) = l(f(3r+1) + 1)2^{3(f(3r+1)+1)}$, for $r \in \mathbb{N}$. Then h belongs to $\mathbf{EXP}(\mathbf{C})$, and by Claim A, $\Omega_M - \Omega_{M_{h(r)}} \leq 2^{-r}$. Hence M is a measure $\mathbf{EXP}(\mathbf{C})$ computable machine.

One finally concludes the proof of Proposition 3.17(a) by classical arguments:

Let $m \in \mathbb{N}$. If $x \in X_m$, then $M(w(m,x)) = x$. Hence $K_M(x) \leq |w(m,x)| = |x| - m + 1$.

Therefore, for $m \geq 1$, $G_m = [X_m] \subseteq R_{m-1}^M$.

(b) Let now $\mathbf{C} = \mathbf{PSPACE}$.

We assume $([X_n])_{n \in \mathbb{N}}$ is an ML- \mathbf{PSPACE} -S test (defined from a set X associated with a machine N). By Remark 3.9, let f be a polynomial function such that for any $n, i \in \mathbb{N}$, X_n is prefix-free, $\mu([X_n]) \leq 2^{-2n}$ and $\mu([X_n]) - \mu([X_{n,f(n+i)}^N]) \leq 2^{-i}$.

Let us start from the bounded request set defined as above, our goal is to define a machine M satisfying properties (3.2) and a polynomial function h such that, for any $r \in \mathbb{N}$,

$$\{w(m,x) : m \leq r, x \in X_{m,f(3r+1)}^{N,\text{space}}\} \subseteq \text{dom}(M_{h(r)}^{\text{space}}). \quad (3.6)$$

If we replace $X_{m,t}^N$ by $X_{m,t}^{N,\text{space}}$ and M_t by M_t^{space} in the previous definitions and claims, the transposed definitions and claims remain valid. Now we must define an algorithm for M with “ $x \in X_{m,f(3r+1)}^{N,\text{space}}$ ” in place of “ $x \in X_{m,f(3r+1)}^N$ ”.

- We use the fact that there is a constant $d \in \mathbb{N}$ such that for any $m, t \in \mathbb{N}$, $X_{m,t}^{N,\text{space}} \subseteq X_{m,2^{dt}}^N$. Hence using $2^{df(3r+1)}$ (under binary representation) as a time-counter, we can check in space $k(f(3r+1))$, for some constant k , whether “ $x \in X_{m,f(3r+1)}^{N,\text{space}}$ ”.

- Also we note that in the previous algorithm, in checking successively for all $(m,x) \in A(r)$ whether $w(m,x) = w$, we needed only to keep track of the value $\rho(m,x)$, for the last browsed (m,x) .

Hence we can define an algorithm for M such that for some polynomial function g , any successful computation of M on w with $|w| = r$ requires at most $g(r)$ cells. Let us set, for $r \in \mathbb{N}$, $h(r) = g(f(3r+1) + 1)$, h satisfies (3.6).

One concludes by the equivalent of Claim A for space, that M is a measure \mathbf{PSPACE} computable machine, and as above that for $m \geq 1$, $G_m = [X_m] \subseteq R_{m-1}^M$.

This concludes the proof of Proposition 3.17. \square

One obtains the opposite direction as an easy generalization of the classical case.

Proposition 3.18.

- (a) Let \mathbf{C} be one of our time-complexity classes. If M is a measure \mathbf{C} computable machine, then $(R_b^M)_{b \in \mathbb{N}}$ (Definition 3.16) is an ML-**EXP**(\mathbf{C})-S test.
- (b) If M is a measure **PSPACE** computable machine, then $(R_b^M)_{b \in \mathbb{N}}$ is an ML-**PSPACE**-S test.

Proof. We refer here to [Nie09, 3.5.14, 3.5.18].

- (a) Let M be a prefix-free machine and let $g \in \mathbf{C}$ be strictly increasing and such that, for any $i \in \mathbb{N}$, $\Omega_M - \Omega_{M_{g(i)}} \leq 2^{-i}$. As in [Nie09, 3.5.15], one shows:

Claim 3.19. For any $x \in \{0, 1\}^*$, $b \in \mathbb{N}$, $K_M(x) \leq |x| - b \Leftrightarrow K_{M_{g(|x|)}}(x) \leq |x| - b$.

Let $X = \{(b, x) \in \mathbb{N} \times \{0, 1\}^* : K_M(x) \leq |x| - b\}$. We consider the machine N which on input (b, x) tests for each $y \in \{0, 1\}^*$ such that $|y| \leq |x| - b$ whether $M_{g(|x|)}(y) = x$. If there is such a y , N halts, otherwise it diverges.

Then by Claim 3.19, $N(b, x) \downarrow \Leftrightarrow (b, x) \in X$. For $b \in \mathbb{N}$, one has $R_b^M = [X_b]$ and classically $\mu([X_b]) \leq 2^{-b}$.

Claim 3.20. $([X_b])_{b \in \mathbb{N}}$ is an ML-**EXP**(\mathbf{C})-S test

Proof. (Sketch referring to [Nie09, 3.5.18]).

By definition of the machine N , there exists an increasing function f in \mathbf{C} such that if N halts on (b, x) , it does so in at most $2^{|x|}f(|x| + b)$ steps. Hence

$$X_b \cap \{x \in \{0, 1\}^* : |x| \leq m\} \subseteq X_{b, 2^m f(m+b)}^N.$$

Let $h \in \mathbf{EXP}(\mathbf{C})$ be such that $h(r) = 2^{g(r)}f(g(r) + r)$. Then

$$X_b \cap \{x \in \{0, 1\}^* : |x| \leq g(m)\} \subseteq X_{b, h(m+b)}^N.$$

Since $M(\sigma) = x$ and $|x| > g(m)$ imply $\sigma \notin \text{dom}(M_{g(m)})$, one then argues classically to deduce

$$\mu([X_b]) - \mu([X_{b, h(m+b)}^N]) \leq 2^{-b}(\Omega_M - \Omega_{M_{g(m)}}) \leq 2^{-m}. \quad \square$$

Let now \mathbf{C} be **PSPACE**. We suppose M is a prefix-free machine, g is a polynomial function such that for any $i \in \mathbb{N}$, $\Omega_M - \Omega_{M_{g(i)}}^{\text{space}} \leq 2^{-i}$.

The set X is defined as in the time-complexity case, but in the rest of the argument, we replace M_t by M_t^{space} . Claim 3.19 can be transposed. On input (b, x) , using a time-counter as previously, the machine N tests in space $p(|x| + b)$, for a polynomial function p , whether there exists $|y| \leq |x| - b$ such that $M_{g(|x|)}^{\text{space}}(y) = x$. Setting $h(r) = p(g(r) + r)$, one concludes as above that $\mu([X_b]) - \mu([X_{b, h(b+m)}^{N, \text{space}}]) \leq 2^{-m}$. This concludes the proof of Proposition 3.18. \square

From Propositions 3.17 and 3.18, we deduce:

Theorem 3.21.

- (a) Let \mathbf{C} be one of our time complexity classes. Then for any $\xi \in \{0, 1\}^{\mathbb{N}}$,
- ξ is ML-**EXP**(\mathbf{C})-S random $\Rightarrow \xi$ is Kolmogorov- \mathbf{C} -S random.
 - ξ is Kolmogorov-**EXP**(\mathbf{C})-S random $\Rightarrow \xi$ is ML- \mathbf{C} -S random.

- (b) Let \mathbf{C} be the class **PSPACE**, **TOWER-EXP** or **PRIM-REC**. Then for any $\xi \in \{0, 1\}^{\mathbb{N}}$, ξ is $ML\text{-}\mathbf{C}$ -S random $\Leftrightarrow \xi$ is Kolmogorov- \mathbf{C} -S random.

3.3. The relation between the Martin-Löf and the martingale notions. Let us deal now with the notion of randomness associated with martingales and orders. We refer to the notion of “inverse” given in Definition 3.10. As a way to obtain true \mathbf{C} orders, let us note:

Claim 3.22.

- (a) If f is an order, then Inv_f is also an order and for $i \geq 1$, $\text{Inv}_{\text{Inv}_f}(i) = f(i - 1) + 1$.
(b) If f is a strictly increasing function in the class \mathbf{C} , where \mathbf{C} is one of our complexity classes, then Inv_f is a true \mathbf{C} order.

Proof.

- (a) Let f be an order. Then for $n \in \mathbb{N}$, $\text{Inv}_f(n + 1) = (\text{least } k \ f(k) \geq n + 1) \geq (\text{least } k \ f(k) \geq n) = \text{Inv}_f(n)$.
For $i \in \mathbb{N}$, $f(i) < f(i) + 1$, hence $\text{Inv}_f(f(i) + 1) > i$. Therefore Inv_f is unbounded.
We deduce that Inv_f is an order.

Now if $h = \text{Inv}_f$, let us compute $\text{Inv}_h(i)$, for $i \geq 1$. We have the equivalences:

$$\begin{aligned} h(k) \geq i &\Leftrightarrow (\text{least } n \ f(n) \geq k) \geq i \\ &\Leftrightarrow f(i - 1) < k. \end{aligned}$$

Therefore, if $i \geq 1$, then $\text{Inv}_h(i) = f(i - 1) + 1$.

- (b) Let f be strictly increasing in \mathbf{C} . We deduce that for $n \in \mathbb{N}$, $f(n) \geq n$. Hence $\text{Inv}_f(n) = \text{least } k \leq n \ f(k) \geq n$. This implies that given our choice of classes \mathbf{C} , Inv_f is also in \mathbf{C} . By (a), Inv_f is an order, and $\text{Inv}_{\text{Inv}_f}$ is in \mathbf{C} . Hence Inv_f is a true \mathbf{C} order. \square

We propose now a result which will be useful in the next section. As an immediate consequence, it shows that when \mathbf{C} is **PSPACE**, requiring the functions from \mathbb{N} to \mathbb{N} - especially orders - to be in \mathbf{P} or in **PSPACE** yields the same notion of randomness.

Claim 3.23.

- (a) Let \mathbf{C} be one of our classes and let f be a true \mathbf{C} order. Then there exists a strictly increasing function g in \mathbf{C} such that $\text{Inv}_g(n) \leq f(n)$ a.e.
(b) Let now f be a true **PSPACE** order. Then there is a true \mathbf{P} order h such that for any $n \in \mathbb{N}$, $h(n) \leq f(n)$.

Proof.

- (a) Let f be a true \mathbf{C} order. We consider the function f' defined by $f'(i) = f(i + 1) \div 1$. Then f' is an order in \mathbf{C} . Also for $n > 0$, we have:

$$\text{Inv}_{f'}(n) = \text{least } k \ (f(k + 1) - 1 \geq n) = \text{Inv}_f(n + 1) \div 1.$$

Hence $\text{Inv}_{f'}$ is a true \mathbf{C} order.

Let us define inductively the function g :

$$\begin{cases} g(0) = \text{inv}_{f'}(0) = 0 \\ g(n + 1) = \max\{\text{Inv}_{f'}(n + 1), g(n) + 1\}. \end{cases}$$

For all our classes \mathbf{C} , g is in \mathbf{C} , it is strictly increasing and satisfies $g \geq \text{Inv}_{f'}$. Hence by definition of Inv , $\text{Inv}_g \leq \text{Inv}_{\text{Inv}_{f'}}$.

Now by Claim 3.22, for $i \geq 1$, $\text{Inv}_g(i) \leq f'(i-1) + 1 = (f(i) \div 1) + 1$.
Let i_0 be least such that $i_0 \geq 1$ and $f(i_0) > 0$. Then for any $i \geq i_0$, $\text{Inv}_g(i) \leq f(i)$.

(b) Let f be a true **PSPACE** order.

We define f' and g from f as in (a). The function g is thus in **PSPACE** and $g \geq \text{Inv}_{f'}$.
Now since integers are under unary representation, there must exist $d, b, k \in \mathbb{N}^*$ such that for any $n \in \mathbb{N}$, $g(n) \leq dn^k + b$. Let $p(n) = dn^k + b$. Then p is strictly increasing and $p \geq g \geq \text{Inv}_{f'}$. We deduce as above $\text{Inv}_p(n) \leq f(n)$ a.e.

By Claim 3.22(b), Inv_p is a true **P** order. \square

Following the terminology of [Lut92] in the polynomial time context, we set:

Definition 3.24. Let D be some class of functions. A martingale $d : \{0, 1\}^* \rightarrow \mathbb{R}^+$ is D -approximable if there exists $F : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}_2$ in D such that for any $i \in \mathbb{N}$, $\|d(x) - F(x, i)\| \leq 2^{-i}$.

The following type of results has already been obtained ([ATZ94, JL95, May94]...).

Lemma 3.25. *Let g be some time-constructible function. If $V : \{0, 1\}^* \rightarrow \mathbb{R}^+$ is a martingale which is $\text{FDTIME}(g(n))$ -approximable, then there exists a \mathbb{Q}_2 -valued martingale d in $\text{FDTIME}(ng(2n+4))$ such that for any $x \in \{0, 1\}^*$, $V(x) \leq d(x) \leq V(x) + 2$.*

We omit the argument (one can adapt [Nie09, 7.3.8] or look up the above references).

To study the relations between ML tests and the martingale-order conditions in the Schnorr subrecursive framework, we shall resort to the following notion from measure theory:

Definition 3.26. Given a measurable subset A of $\{0, 1\}^{\mathbb{N}}$ and $x \in \{0, 1\}^*$, the conditional measure $\mu(A|x)$ is the quotient $\frac{\mu(A \cap [x])}{\mu([x])} = 2^{|x|} \mu(A \cap [x])$.

Classically, the function $d : x \mapsto \mu(A|x)$ is a martingale.

Proposition 3.27.

- (a) *Let \mathbf{C} be one of our time-complexity classes. If $(G_n)_{n \in \mathbb{N}}$ is an ML- \mathbf{C} -S test, then there exist an **EXP**(\mathbf{C})-approximable martingale B and a true \mathbf{C} order h such that*

$$\text{for any } \xi \in \{0, 1\}^{\mathbb{N}}, \xi \in \bigcap_{n \in \mathbb{N}} G_n \Rightarrow B(\xi \upharpoonright i) \geq 2^{h(i)} \quad \text{i.o.}$$
- (b) *Given an ML-**PSPACE**-S test $(G_n)_{n \in \mathbb{N}}$, there exist a **PSPACE**-approximable martingale B and a true **PSPACE** order h satisfying the above implication for any $\xi \in \{0, 1\}^{\mathbb{N}}$.*

Proof.

- (a) Let \mathbf{C} be one of our time-complexity class and let $(G_n)_{n \in \mathbb{N}}$ be an ML- \mathbf{C} -S test. We can assume there are $X \subseteq \mathbb{N} \times \{0, 1\}^*$, a machine M and a function $f : \mathbb{N} \rightarrow \mathbb{N}$ strictly increasing in \mathbf{C} such that

- $X = \{(n, x) \in \mathbb{N} \times \{0, 1\}^* : M(n, x) \downarrow\}$,
- for $n \in \mathbb{N}$, $X_n = \{x \in \{0, 1\}^* : (n, x) \in X\}$ is prefix-free, $G_n = [X_n]$ and $\mu(G_n) \leq 2^{-2n}$,
- for $n, i \in \mathbb{N}$, $\mu([X_n]) - \mu([X_{n, f(n+i)}^M]) \leq 2^{-i}$.

Definition 3.28.

- let $g : \mathbb{N} \rightarrow \mathbb{N}$ be such that, for $i \in \mathbb{N}$, $g(i) = f(5i)$.
- For $n, k \in \mathbb{N}$, let $C_n^k = [X_n \setminus X_{n, g(k)}^M]$.
- Define $B : \{0, 1\}^* \rightarrow \mathbb{R}^+$ as follows: for $x \in \{0, 1\}^*$, $B(x) = \sum_{n, k} 2^k \mu(C_n^k|x)$.

For $n, k \in \mathbb{N}$, since X_n is prefix-free, we have $[X_n \setminus X_{n,g(k)}^M] = [X_n] \setminus [X_{n,g(k)}^M]$. Let us note that for $n \leq 2k$,

$$\mu(C_n^k) = \mu([X_n]) - \mu([X_{n,g(k)}^M]) \leq \mu([X_n]) - \mu([X_{n,f(n+3k)}^M]) \leq 2^{-3k}. \quad (3.7)$$

Hence for any $n \in \mathbb{N}$,

$$\mu(C_n^k) \leq \mu(G_n) \leq 2^{-2n}. \quad (3.8)$$

Claim 3.29. B is a martingale.

Proof. We only need to check that $B(\emptyset)$ is finite. One has

$$\begin{aligned} B(\emptyset) &= \sum_k \sum_{n \leq 2k} 2^k \mu(C_n^k) + \sum_k \sum_{n > 2k} 2^k \mu(C_n^k) \\ &\leq \sum_k \sum_{n \leq 2k} 2^k 2^{-3k} + \sum_k \sum_{n > 2k} 2^k 2^{-n} \quad (\text{by (3.7) and (3.8)}) \\ &\leq \sum_k 2k 2^{-2k} + \sum_k 2^k 2^{-2k} \\ &\leq 6 \quad (\text{by } k \leq 2^k). \end{aligned} \quad \square$$

Now let us show the following:

Claim 3.30. The function $h = \text{Inv}_g \dot{-} 1$ is a true \mathbf{C} order, and for any $\xi \in \{0, 1\}^{\mathbb{N}}$, $\xi \in \bigcap_{n \in \mathbb{N}} G_n \Rightarrow B(\xi \upharpoonright i) \geq 2^{h(i)}$ i.o.

Proof. If $h = \text{Inv}_g \dot{-} 1$, then for $n \geq 1$, $\text{Inv}_h(n) = \text{Inv}_{\text{Inv}_g}(n+1)$. Hence by Claim 3.22(b), h is a true \mathbf{C} order. Now for the second statement, we provide an argument for our precise definition of the martingale B , but the line of proof should be as in Schnorr's original demonstration.

Let $\xi \in \bigcap_{n \in \mathbb{N}} G_n$. For any $n \in \mathbb{N}$, there must exist $i_n \in \mathbb{N}$ such that $\xi \upharpoonright i_n \in X_n$. The inclusion $[\xi \upharpoonright i_n] \subseteq G_n$ implies $2^{-i_n} \leq \mu(G_n) \leq 2^{-2n}$. Hence for any $n \in \mathbb{N}$, $i_n \geq n$. We check

$$\text{for any } n > g(0), B(\xi \upharpoonright i_n) \geq 2^{h(i_n)}. \quad (3.9)$$

We now assume $n > g(0)$, then also $i_n > g(0)$ and hence $\text{Inv}_g(i_n) \geq 1$. By definition of Inv , $g(\text{Inv}_g(i_n) - 1) < i_n$. Let $k_n = h(i_n) = \text{Inv}_g(i_n) - 1$. Since $g(k_n) < i_n$, necessarily $\xi \upharpoonright i_n \notin X_{n,g(k_n)}^M$. We know $\xi \upharpoonright i_n \in X_n$, hence necessarily $\xi \upharpoonright i_n \in X_n \setminus X_{n,g(k_n)}^M$. This gives $\mu(C_n^{k_n} | \xi \upharpoonright i_n) = 1$.

Therefore $B(\xi \upharpoonright i_n) = \sum_{k,n} 2^k \mu(C_n^k | \xi \upharpoonright i_n) \geq 2^{k_n} = 2^{h(i_n)}$. \square

Our goal now is to find a function $F : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}_2$ which **EXP(C)**-approximates B . Before developing the whole proof, we summarize the argument:

- for $x \in \{0, 1\}^*$, $B(x)$ is an infinite sum of real terms $2^k \mu(C_n^k | x)$. We first truncate $B(x)$ to obtain a finite sum $B_2(x, i) = \sum_{(n,k) \in A(|x|, i)} 2^k \mu(C_n^k | x)$, for $A(|x|, i)$ finite $\subseteq \mathbb{N} \times \mathbb{N}$ appropriately bounded so that $B_2(x, i)$ approximates $B(x)$ within $2^{-(i-1)}$.
- The second step consists in replacing, in the finite sum $B_2(x, i)$, each term $2^k \mu(C_n^k | x) = 2^k 2^{|x|} \mu([X_n \setminus X_{n,g(k)}^M] \cap [x])$ by the term $2^k 2^{|x|} \mu([X_{n,\bar{g}(|x|,i)}^M \setminus X_{n,g(k)}^M] \cap [x])$ for an adequate function \bar{g} in \mathbf{C} .

By switching from measures of open sets $[X_n \setminus X_{n,g(k)}^M] \cap [x]$ to measures of clopen sets

$[X_{n,\bar{g}(|x|,i)}^M \setminus X_{n,g(k)}^M] \cap [x]$, we shall obtain a sum $F(x, i) = \sum_{(n,k) \in A(|x|,i)} 2^k \mu([X_{n,\bar{g}(|x|,i)}^M \setminus X_{n,g(k)}^M] \cap [x])$ in \mathbb{Q}_2 with the expected approximation properties.

- Moreover the bounds on $A(|x|, i)$ (polynomial in $(|x|, i)$) and the fact that \bar{g} belongs to \mathbf{C} will imply that $F : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}_2$ belongs to $\mathbf{EXP}(\mathbf{C})$.

Hence let $B(x) = \sum_{k,n} 2^k \mu(C_n^k | x)$.

- (i) We first bind the integer n in the sum. Having set $N(r, i, k) = r + i + 2k + 3$, we consider

$$B_1(x, i) = \sum_k \sum_{n \leq N(|x|, i, k)} 2^k \mu(C_n^k | x).$$

Then

$$\begin{aligned} 0 \leq B(x) - B_1(x, i) &= \sum_k \sum_n 2^k \mu(C_n^k | x) - \sum_k \sum_{n \leq N(|x|, i, k)} 2^k \mu(C_n^k | x), \\ &= \sum_k \sum_{n > N(|x|, i, k)} 2^k \mu(C_n^k | x), \\ &\leq \sum_k 2^k 2^{|x|} \sum_{n > N(|x|, i, k)} \mu(G_n), \\ &\leq 2^{|x|} \sum_k 2^k 2^{-N(|x|, i, k)} \quad (\text{by } \mu(G_n) \leq 2^{-n}) \\ &\leq 2^{-i-2}. \end{aligned} \tag{3.10}$$

- (ii) Let us deal now with k . We set $K(r, i) = r + i + 4$ and consider

$$B_2(x, i) = \sum_{k \leq K(|x|, i)} \sum_{n \leq N(|x|, i, k)} 2^k \mu(C_n^k | x).$$

Our goal is to show $0 \leq B_1(x, i) - B_2(x, i) \leq 2^{-i-2}$.

Let us set $\varepsilon_k(x, i) = \sum_{n \leq N(|x|, i, k)} 2^k \mu(C_n^k | x)$.

Claim 3.31. If $k > |x| + i + 4$, then $\varepsilon_k(x, i) \leq 2^{|x|-k+2}$.

Proof.

– If $n \leq 2k$, then by (3.7), $\mu(C_n^k \cap [x]) \leq 2^{-3k}$.

– If $n > 2k$, then by (3.8), $\mu(C_n^k \cap [x]) \leq 2^{-2n} < 2^{-4k}$.

Hence for all $k, n \in \mathbb{N}$, $\mu(C_n^k \cap [x]) \leq 2^{-3k}$.

The hypothesis $k > |x| + i + 4$ implies $N(|x|, i, k) = |x| + i + 2k + 3 \leq 3k$. Hence we deduce that for any $k \in \mathbb{N}$, $\varepsilon_k(x, i) \leq N(|x|, i, k) 2^k 2^{-3k} 2^{|x|} \leq 3k 2^{-2k+|x|} \leq 2^{2+|x|-k}$. \square

Therefore

$$0 \leq B_1(x, i) - B_2(x, i) = \sum_{k > |x| + i + 4} \varepsilon_k(x, i) \leq \sum_{k > |x| + i + 4} 2^{2+|x|-k} \leq 2^{-i-2}. \tag{3.11}$$

- (iii) We now define the function $\bar{g} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ used to switch from the open sets C_n^k to clopen sets.

Definition 3.32. Let $\bar{g}(r, i) = f(9r + 9i + 32)$ and $D_n^{k,i,r} = [X_{n,\bar{g}(r,i)}^k \setminus X_{n,g(k)}^k]$. We set

$$F(x, i) = \sum_{k \leq K(|x|, i)} \sum_{n \leq N(|x|, i, k)} 2^k \mu(D_n^{k,i,|x|} | x).$$

Our goal is to obtain $0 \leq B_2(x, i) - F(x, i) \leq 2^{-i-1}$.

- We first check that for $k \leq K(r, i)$, $\bar{g}(r, i) \geq g(k)$:
 $\bar{g}(r, i) = f(9r + 9i + 32) \geq f(5(r + i + 4)) = f(5K(r, i)) \geq f(5k) = g(k)$.
- Let us check that for $k \leq K(r, i)$ and $n \leq N(r, i, k)$, $\bar{g}(r, i) \geq f(n + (2r + 2i + 4k + 6))$. One computes $N(r, i, K(r, i)) = 3r + 3i + 11$. Hence

$$\begin{pmatrix} k \leq K(r, i) \\ n \leq N(r, i, k) \end{pmatrix} \Rightarrow \begin{pmatrix} k \leq r + i + 4 \\ n \leq 3r + 3i + 11 \end{pmatrix} \Rightarrow n + (2r + 2i + 4k + 5) \leq 9r + 9i + 32.$$

One deduces

$$f(n + (2r + 2i + 4k + 5)) \leq f(9r + 9i + 32) \leq \bar{g}(r, i). \quad (3.12)$$

- For $k \leq K(|x|, i)$ and $n \leq N(|x|, i, k)$, let us set $\Delta(n, k, x, i) = \mu(C_n^k \cap [x]) - \mu(D_n^{k, i, |x|} \cap [x])$. One obtains:

$$\begin{aligned} \Delta(n, k, x, i) &= \mu((C_n^k \setminus D_n^{k, i, |x|}) \cap [x]) \\ &\leq \mu(C_n^k \setminus D_n^{k, i, |x|}) \\ &\leq \mu([(X_n \setminus X_{n, g(k)}^M) \setminus (X_{n, \bar{g}(|x|, i)}^M \setminus X_{n, g(k)}^M)]) \\ &\leq \mu([X_n \setminus X_{n, \bar{g}(|x|, i)}^M]) \\ &\leq \mu([X_n \setminus X_{n, f(n + (2|x| + 2i + 4k + 5))}^M]) \text{ (by (3.12))} \\ &\leq 2^{-(2|x| + 2i + 4k + 5)} \end{aligned} \quad (3.13)$$

We derive:

$$\begin{aligned} 0 \leq B_2(x, i) - F(x, i) &= \sum_{k \leq K(|x|, i)} \sum_{n \leq N(|x|, i, k)} 2^k 2^{|x|} \mu((C_n^k \setminus D_n^{k, i, |x|}) \cap [x]) \\ &= \sum_{k \leq K(|x|, i)} \sum_{n \leq N(|x|, i, k)} 2^{k+|x|} \Delta(n, k, x, i) \\ &\leq \sum_{k \leq K(|x|, i)} \sum_{n \leq N(|x|, i, k)} 2^{k+|x|} 2^{-(2|x| + 2i + 4k + 5)} \quad \text{(by (3.13))} \\ &\leq \sum_{k \leq K(|x|, i)} N(|x|, i, k) 2^{-(|x| + 2i + 3k + 5)} \\ &\leq \sum_k 2^{|x| + i + 2k + 3} 2^{-(|x| + 2i + 3k + 5)} \\ &\leq 2^{-i-1}. \end{aligned} \quad (3.14)$$

Combining (3.10), (3.11) and (3.14), we deduce:

Claim 3.33. For any $x \in \{0, 1\}^*$ and $i \in \mathbb{N}$, $0 \leq B(x) - F(x, i) \leq 2^{-i}$.

It now remains to evaluate the complexity of F . Clearly if f is in \mathbf{C} , then \bar{g} is also in \mathbf{C} . Given k, n, i, x , to compute $\mu(D_n^{k, i, |x|} \cap [x])$, one has to check for each finite sequence z of length $\leq \bar{g}(|x|, i)$ compatible with x whether it belongs to $X_{n, \bar{g}(|x|, i)}^M \setminus X_{n, g(k)}^M$ and to compare it with x .

- If there is $z \in X_{n, \bar{g}(|x|, i)}^M \setminus X_{n, g(k)}^M$ such that $z \preceq x$, then $\mu(D_n^{k, i, |x|} \cap [x])$ is $2^{-|x|}$.

- Otherwise one adds all $2^{-|z|}$, for $x \prec z$ with z in $X_{n,\bar{g}(|x|,i)}^M \setminus X_{n,g(k)}^M$ to obtain the measure. All intermediate (and the final) sums can be coded, for some constant d , by strings of length $\leq d(\bar{g}(|x|,i))$ (the total measure is ≤ 1).

Hence the function $\varphi : \mathbb{N}^3 \times \{0,1\}^* \rightarrow \mathbb{Q}_2$ is in **EXP(C)**.
 $(k,n,i,x) \mapsto \mu(D_n^{k,i,|x|} \cap [x])$

Since for some constant c , our bounds $K(|x|,i)$ and $N(|x|,i,k)$ are $\leq c(|x|+i)$, we deduce

Claim 3.34. F belongs to the class **EXP(C)**.

We can now conclude the proof of (a) by Claims 3.29, 3.30, 3.33 and 3.34.

Let now **C** be the class **PSPACE**.

We assume the sequence $([X_n])_{n \in \mathbb{N}}$ associated with a machine M , satisfies for some (strictly increasing) polynomial function f :

- $\mu([X_n]) \leq 2^{-2n}$,
- $\mu([X_n]) - \mu([X_{n,f(n+i)}^{M,\text{space}}]) \leq 2^{-i}$.

One defines g, h and \bar{g} from f as in (a). They are all polynomial functions. C_n^k is now the open set $[X_n \setminus X_{n,g(k)}^{M,\text{space}}]$ and one also considers the martingale $B(x) = \sum_{n,k} 2^k \mu(C_n^k | x)$.

(Since $|x| > s$ implies $x \notin X_{n,s}^{M,\text{space}}$) one obtains the equivalent of Claim 3.30: h is a true **PSPACE** order and for any $\xi \in \{0,1\}^{\mathbb{N}}$,

$$\xi \in \bigcap_{n \in \mathbb{N}} G_n \Rightarrow B(\xi \upharpoonright i) \geq 2^{h(i)} \text{ i.o.}$$

We set $D_n^{k,i,r} = [X_{n,\bar{g}(r,i)}^{M,\text{space}} \setminus X_{n,g(k)}^{M,\text{space}}]$ and define the approximating function F as in (a) with the $D_n^{k,i,r}$'s.

To compute $\mu(D_n^{k,i,|x|} \cap [x])$, we also enumerate all sequences z of length $\leq \bar{g}(|x|,i)$ according to \preceq_{lex} , and (using counters) we test whether $z \in D_n^{k,i,|x|}$. But this time we only keep track of the last browsed sequence z and of the partial measure $\mu([(X_{n,\bar{g}(|x|,i)} \setminus X_{n,g(k)}) \cap \{t : t \preceq_{\text{lex}} s\}] \cap [x])$. As in (a), we know this partial measure is coded by a string of length $\mathcal{O}(\bar{g}(|x|,i))$.

Hence $\varphi : \mathbb{N}^3 \times \{0,1\}^* \rightarrow \mathbb{Q}_2$ is in **PSPACE**.
 $(k,n,i,x) \mapsto \mu(D_n^{k,i,|x|} \cap [x])$

We deduce that F is in **PSPACE** and conclude the proof of (b) as above. \square

By Proposition 3.27 and lemma 3.25 about approximation, we derive:

Proposition 3.35.

- Let **C** be one of our time-complexity classes. If $(G_n)_{n \in \mathbb{N}}$ is an **ML-C-S** test, then there exist a martingale $d : \{0,1\}^* \rightarrow \mathbb{Q}_2$ in **EXP(C)** and a true **C** order h such that for any $\xi \in \{0,1\}^{\mathbb{N}}$, $\xi \in \bigcap_{n \in \mathbb{N}} G_n \Rightarrow B(\xi \upharpoonright i) \geq 2^{h(i)}$ i.o.
- Given an **ML-PSPACE-S** test $(G_n)_{n \in \mathbb{N}}$, there exist a martingale $d : \{0,1\}^* \rightarrow \mathbb{Q}_2$ in **PSPACE** and a true **PSPACE** order h satisfying the above implication for any $\xi \in \{0,1\}^{\mathbb{N}}$.

The opposite direction - from martingales to Martin-Löf tests - is easier and can be obtained through a simple adaptation of existing arguments.

Proposition 3.36.

- (a) Let \mathbf{C} be one of our time-complexity classes. From a martingale $d : \{0, 1\}^* \rightarrow \mathbb{Q}_2$ in \mathbf{C} and a true \mathbf{C} order g , one can define an ML- \mathbf{C} -S test $(G_n)_{n \in \mathbb{N}}$ such that
for any $\xi \in \{0, 1\}^{\mathbb{N}}$, $d(\xi \upharpoonright i) \geq 2^{g(i)}$ i.o. $\Rightarrow \xi \in \bigcap_{n \in \mathbb{N}} G_n$.
- (b) Given a **PSPACE** martingale d and a true **PSPACE** order g , one can construct an ML-**PSPACE**-S test $(G_n)_{n \in \mathbb{N}}$, satisfying for any $\xi \in \{0, 1\}^{\mathbb{N}}$, the above implication.

Proof. Let d and g be respectively the martingale and the order. We can assume $d(\emptyset) \leq 1$.

Let us consider the set $X = \{(n, x) \in \mathbb{N} \times \{0, 1\}^* : d(x) \geq 2^{g(|x|)} \geq 2^n\}$, and for $n \in \mathbb{N}$, the associated set $X_n = \{x \in \{0, 1\}^* : (n, x) \in X\}$.

Setting for $n \in \mathbb{N}$, $G_n = [X_n]$, one obtains by classical arguments ([DH10, 7.1.7], [Nie09, 7.3.3]) that $(G_n)_{n \in \mathbb{N}}$ is an ML test and that for any $\xi \in \{0, 1\}^{\mathbb{N}}$,

$$d(\xi \upharpoonright i) \geq 2^{g(i)} \text{ i.o.} \Rightarrow \xi \in \bigcap_{n \in \mathbb{N}} G_n.$$

We must now check that $(G_n)_{n \in \mathbb{N}}$ is an ML- \mathbf{C} -S test. We shall make explicit the use of Inv in Schnorr's original proof and add a few lines to define the controlling function.

- (a) Let \mathbf{C} be one of our time-complexity classes. To deal with prefix-free sets, we consider minimal strings for \prec :

$$\begin{aligned} \text{Let } Y &= \{(n, x) \in X : \forall y \prec x \ (n, y) \notin X\} \\ &= \{(n, x) \in \mathbb{N} \times \{0, 1\}^* : d(x) \geq 2^{g(|x|)} \geq 2^n \wedge \forall y \prec x \ \neg(d(y) \geq 2^{g(|y|)} \geq 2^n)\}. \end{aligned}$$

For some constant d and some function $f' \in F_C$, membership of (n, x) in Y can be checked in time $\leq f'(|x| + n)$. Hence one can define a machine M and a strictly increasing function $f \in \mathbf{C}$ such that for any $(n, x) \in \mathbb{N} \times \{0, 1\}^*$,

$$\begin{aligned} (n, x) \in Y &\Leftrightarrow M(n, x) \downarrow \\ &\Leftrightarrow (n, x) \in \text{dom}(M_{f(|x|+n)}). \end{aligned}$$

Setting, for $n \in \mathbb{N}$, $Y_n = \{x \in \{0, 1\}^* : (n, x) \in Y\}$. Then $[X_n] = [Y_n]$ and we get, for $n \in \mathbb{N}$, $x \in \{0, 1\}^*$,

$$x \in Y_n \Leftrightarrow x \in Y_{n, f(|x|+n)}^M.$$

For all $n, k \in \mathbb{N}$, one has:

$$Y_n \cap \{x \in \{0, 1\}^* : |x| \leq \text{Inv}_g(k)\} \subseteq Y_{n, f(\text{Inv}_g(k)+n)}. \quad (3.15)$$

For any $m \in \mathbb{N}$, $g(\text{Inv}_g(m)) \geq m$, hence

$$\begin{aligned} Y_n \cap \{x \in \{0, 1\}^* : |x| > \text{Inv}_g(k)\} &\subseteq \{x \in \{0, 1\}^* : d(x) \geq 2^{g(|x|)} \geq 2^{g(\text{Inv}_g(k))}\} \\ &\subseteq \{x \in \{0, 1\}^* : d(x) \geq 2^k\}. \end{aligned} \quad (3.16)$$

Now by [DH10, 6.3.3], [Nie09, 7.1.9], since $d(\emptyset) \leq 1$,

$$\mu(\{x \in \{0, 1\}^* : d(x) \geq 2^k\}) \leq 2^{-k}. \quad (3.17)$$

Let us set $\bar{g}(r) = f(\text{Inv}_g(r) + r)$, for $r \in \mathbb{N}$. Since g is a true \mathbf{C} order, \bar{g} is in \mathbf{C} . For any $k, n \in \mathbb{N}$, $\bar{g}(k+n) \geq f(\text{Inv}_g(k) + n)$, we thus deduce:

$$\begin{aligned} \mu([Y_n]) - \mu([Y_{n, \bar{g}(k+n)}]) &= \mu([Y_n \setminus Y_{n, \bar{g}(k+n)}]) \\ &\leq \mu([Y_n \cap \{x \in \{0, 1\}^* : |x| > \text{Inv}_g(k)\}]) \text{ (by (3.15))} \\ &\leq 2^{-k} \text{ (by (3.16) and (3.17)).} \end{aligned}$$

Therefore $(G_n)_{n \in \mathbb{N}} = ([X_n])_{n \in \mathbb{N}} = ([Y_n])_{n \in \mathbb{N}}$ is an ML- \mathbf{C} -S test.

- (b) Let now \mathbf{C} be **PSPACE** and let d in **PSPACE** and g a true **PSPACE** order satisfy the implication. The set Y being defined as above, membership in Y can be tested in polynomial space (either by using a time-counter in binary or by Claim 3.23). Hence there exist a machine M and a polynomial function f so that for any $(n, x) \in \mathbb{N} \times \{0, 1\}^*$,

$$\begin{aligned} (n, x) \in Y &\Leftrightarrow M(n, x) \downarrow \\ &\Leftrightarrow (n, x) \in \text{dom}(M_{f(|x|+n)}^{\text{space}}). \end{aligned}$$

One then concludes as above (with M_s^{space} and $Y_{n,s}^{M,\text{space}}$ instead of M_s and $Y_{n,s}^M$). \square

Remark 3.37. Replacing the classical requirement “ $d(\xi \upharpoonright_i) \geq g(i)$ i.o” by “ $d(\xi \upharpoonright_i) \geq 2^{g(i)}$ i.o” allowed us to consider the function $\bar{g}(r) = f(\text{Inv}_g(r) + r)$ instead of the function $f(\text{Inv}_g(2^r) + r)$. This was essential to get Proposition 3.36 for $\mathbf{C} = \mathbf{P}, \mathbf{EXP}$ or **PSPACE**.

Combining Propositions 3.35 and 3.36, we deduce:

Theorem 3.38.

- (a) Let \mathbf{C} be one of our time-complexity classes. Then for any $\xi \in \{0, 1\}^{\mathbb{N}}$,
 ξ is martingale-**EXP**(\mathbf{C})-S random $\Rightarrow \xi$ is ML- \mathbf{C} -S random $\Rightarrow \xi$ is martingale- \mathbf{C} -S random.
- (b) Let \mathbf{C} be the class **PSPACE**, **TOWER-EXP** or **PRIM-REC**. Then for any sequence $\xi \in \{0, 1\}^{\mathbb{N}}$,
 ξ is martingale- \mathbf{C} -S random $\Leftrightarrow \xi$ is ML- \mathbf{C} -S random.

Finally merging theorems 3.21 and 3.38, we obtain:

Theorem 3.39. Let \mathbf{C} be the class **PSPACE**, **TOWER-EXP** or **PRIM-REC**. Then for any $\xi \in \{0, 1\}^{\mathbb{N}}$,
 ξ is ML- \mathbf{C} -S random $\Leftrightarrow \xi$ is Kolmogorov - \mathbf{C} -S random $\Leftrightarrow \xi$ is martingale- \mathbf{C} -S random.

4. SEPARATION.

To justify our previous work, we now differentiate Schnorr randomness from (martingale)-**PRIM-REC**-S randomness by appealing to the following notion:

Definition 4.1. Let \mathbf{C} be a class of functions. An infinite binary sequence ξ is \mathbf{C} random if any martingale $d : \{0, 1\}^* \rightarrow \mathbb{Q}_2$ in \mathbf{C} fails on ξ (i.e. the set $\{d(\xi \upharpoonright_i) : i \in \mathbb{N}\}$ is bounded in \mathbb{N}).

If \mathbf{C} is the class of computable functions, then the above notion is “computable randomness” (Schnorr). When \mathbf{C} is \mathbf{P} , this is “p-randomness” (Lutz).

The following argument was suggested by one of the (anonymous) referees: let $A : \mathbb{N} \rightarrow \mathbb{N}$ be a computable function dominating all primitive recursive functions. Then by classical results (see [AM97, 3.9.7] for a precise statement), there exists a computable sequence $\xi \in \{0, 1\}^{\mathbb{N}}$ which is $\text{FDTIME}(A(n))$ random. ξ is thus **PRIM-REC**-random and hence (martingale)-**PRIM-REC**-S random. Therefore Schnorr’s randomness is strictly stronger than (martingale)**PRIM-REC**-S randomness.

Our original argument was based on the notion of ML-**PRIM-REC**-S randomness. The method - though laborious - could be extended to prove the assertions:

- ML-**PRIM-REC**-S randomness $>$ ML-**TOWER-EXP**-S randomness and
- ML-**TOWER-EXP**-S randomness $>$ ML-**EXP**-S randomness.

But we could not deduce that ML-**EXP**-S randomness is strictly stronger than ML-**P**-S randomness whereas this can be done for the martingale corresponding notion of S randomness. The martingale approach seems better suited to low time-complexity classes, we shall thus build on the important amount of work developed around the notion of martingale in the field of Resource Bounded Randomness.

In this section, we shall restrict ourselves to time-complexity classes and we shall focus on the martingale definition of S-randomness. For such a class \mathbf{C} , the expression “martingale- \mathbf{C} -S random” will be abbreviated to “ \mathbf{C} -S random”. As \mathbf{C} rises among our time-complexity classes, the notion of \mathbf{C} -S randomness gets strictly stronger. We shall compare our notion of \mathbf{P} -S randomness with Lutz notion of p-randomness, Wang’s notion of (\mathbf{P}, \mathbf{P}) -S randomness and we shall also contrast the notion of **PRIM-REC**-S randomness with the notion of BP-randomness developed by Buss, Cenzer and Remmel.

4.1. \mathbf{C} -S randomness and \mathbf{C} randomness. Let us mention first the work of Wang [Wan99, Wan00] who studied a version of Schnorr randomness for the class \mathbf{P} (termed (\mathbf{P}, \mathbf{P}) -S randomness) and proved it to be weaker than the notion of p-randomness [Wan00, Thm 8].

Definition 4.2 ([Wan00]). Let \mathbf{C} be a class of functions. An infinite sequence ξ is (\mathbf{C}, \mathbf{C}) -S random iff for any martingale F and any order h both in \mathbf{C} , $F(\xi \upharpoonright i) < h(i)$ a.e.

His notion is stronger than ours because he allows all orders in \mathbf{C} , not restricting to true \mathbf{C} orders. (If one is not concerned with the status of the inverse of the order, our condition “ $d(\xi \upharpoonright i) < 2^{h(i)}$ a.e.” and the classical one “ $d(\xi \upharpoonright i) < h(i)$ a.e.” yield the same notion of randomness for our classes \mathbf{C}). A consequence of his definition is that for computable infinite sequences, p-randomness and (\mathbf{P}, \mathbf{P}) -S randomness coincide [Wan00, Cor. 17].

Building on his results and techniques, we shall show that our definition allows more variety inside the set of computable infinite sequences.

Let \mathbf{C} be one of our time-complexity classes. To separate \mathbf{C} randomness from \mathbf{C} -S randomness inside the set of computable sequences (and to obtain the tableau of the introduction) we shall rely on part (i) of the following proposition; the remaining cases (ii)-(iv) add precision, showing that the sequence which is \mathbf{C} -S random but not \mathbf{C} random can be taken “right above \mathbf{C} ”.

For $\xi \in \{0, 1\}^{\mathbb{N}}$ and $g \in \mathbb{N}^{\mathbb{N}}$ time-constructible, we say that ξ belongs to $FDTIME(g(n))$ if the function $n \mapsto \xi(n)$ belongs to $FDTIME(g(n))$ (n under unary representation).

Proposition 4.3.

- (i) *There is a computable $\xi \in \{0, 1\}^{\mathbb{N}}$ which is **PRIM-REC**-S random but not \mathbf{P} random.*
- (ii) *There exists $\xi \in \{0, 1\}^{\mathbb{N}}$ in $FDTIME(n^{\lceil \log n \rceil})$ which is \mathbf{P} -S random but not \mathbf{P} random.*
- (iii) *There is $\xi \in \{0, 1\}^{\mathbb{N}}$ in $FDTIME(2^{n^{\lceil \log n \rceil}})$ which is **EXP**-S random but not \mathbf{P} random.*
- (iv) *There is $\xi \in \{0, 1\}^{\mathbb{N}}$ in $FDTIME(T(\lceil \log n \rceil, n))$ which is **TOWER-EXP**-S random but not \mathbf{P} random.*

Our proof will be based on enumerations of martingales and of true **PRIM-REC** orders. We thus propose without proof a few definitions and classical (or easy) facts:

Definition 4.4. Let \mathbf{C} be one of our time-complexity classes. We assume $G_C : \mathbb{N} \times \{0, 1\}^* \rightarrow \mathbb{Q}_2$ enumerates all functions $f : \{0, 1\}^* \rightarrow \mathbb{Q}_2$ in \mathbf{C} , and $g_C : \mathbb{N} \rightarrow \mathbb{N}$ strictly increasing is such that $G_C \in \text{FDTIME}(g_C(n))$.

- (a) Then there is $d_C : \mathbb{N} \times \{0, 1\}^* \rightarrow \mathbb{Q}_2$ enumerating all martingales d in \mathbf{C} with $d(\emptyset) \leq 1$ which is such that $d_C \in \text{FDTIME}(ng_C(n))$ (see for example [ATZ94]).
- (b) Let us define the martingale $\Phi_C(x) = \sum_{e \in \mathbb{N}} 2^{-e} d_C((e)_0, x)$. It can be approximated by $f_C : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}_2$ defined by $f_C(x, i) = \sum_{e \leq i + |x|} 2^{-e} d_C((e)_0, x)$ which belongs to $\text{FDTIME}(n^2 g_C(2n))$.
- (c) By Lemma 3.25, there is a \mathbb{Q}_2 -valued martingale δ_C in $\text{FDTIME}(n^3 g_C(5n))$ such that for any $x \in \{0, 1\}^*$, $\Phi_C(x) \leq \delta_C(x) \leq \Phi_C(x) + 2$.

The notation $(e)_0$ or $(e)_1$ refers to the inverses of the polynomial time bijection from $\mathbb{N} \times \mathbb{N}$ onto \mathbb{N} . If \mathbf{C} is **PRIM-REC**, then both G_C and g_C can be taken recursive. Hence in (c), we only assert that $\delta_{\mathbf{PRIM-REC}}$ is recursive. In all cases, g_C will be time-constructible. Here are some possible choices for $(G_C$ and) g_C :

Fact 4.5. One can take

- (i) $g_{\mathbf{PRIM-REC}}$ recursive
- (ii) $g_{\mathbf{P}}(n) = n^{\lceil (2/3) \log n \rceil}$
- (iii) $g_{\mathbf{EXP}}(n) = 2^{\lceil (1/2) \log n \rceil}$
- (iv) $g_{\mathbf{TOWER-EXP}}(n) = T(\lceil (1/2) \log n \rceil, n)$.

Proof. We give a few details for (ii), the other cases are very similar.

We want to define $G_{\mathbf{P}} : \mathbb{N} \times \{0, 1\}^* \rightarrow \mathbb{Q}_2$ enumerating all polynomial time functions from $\{0, 1\}^*$ into \mathbb{Q}_2 . Let M be a universal machine which for some constant $c \in \mathbb{N}$, simulates the computation of $t(n)$ steps of the machine M_e (with program $e \in \mathbb{N}$) in $ce t(n) \lceil \log t(n) \rceil$ steps. We define an algorithm for $G_{\mathbf{P}}$: On input (e, x) , one computes $(e)_0$, $(e)_1$ and $\lceil (1/2) \log(e)_1 \rceil$. Then M simulates $M_{(e)_0}$ on x during $|x|^{\lceil (1/2) \log(e)_1 \rceil}$ steps. If $M_{(e)_0}$ halts, then $G_{\mathbf{P}}$ outputs the result of the computation, otherwise it outputs 0.

One can check that $G_{\mathbf{P}}$ enumerates all polynomial time functions and that the function $G_{\mathbf{P}}$ belongs to $\text{FDTIME}(n^{\lceil (2/3) \log n \rceil})$ (the input (e, x) has size $n = e + |x|$). \square

Referring to the martingale $\delta_{\mathbf{C}}$ in Definition 4.4, one derives from the previous values:

Fact 4.6.

- (i) $\delta_{\mathbf{PRIM-REC}}$ is recursive
- (ii) $\delta_{\mathbf{P}} \in \text{FDTIME}(n^{\lceil (3/4) \log n \rceil})$
- (iii) $\delta_{\mathbf{EXP}} \in \text{FDTIME}(2^{n^{\lceil (3/4) \log n \rceil}})$
- (iv) $\delta_{\mathbf{TOWER-EXP}} \in \text{FDTIME}(T(\lceil (3/4) \log n \rceil, n))$

Let $(\varphi_e)_{e \in \mathbb{N}}$ be the usual effective enumeration of partial recursive functions from (a subset of) \mathbb{N} into \mathbb{N} . We obtain the following:

Lemma 4.7. *There exists a (total) recursive function $l \in \mathbb{N}^{\mathbb{N}}$ such that $(\varphi_{l(e)})_{e \in \mathbb{N}}$ is an enumeration of all inverses of strictly increasing primitive recursive functions in $\mathbb{N}^{\mathbb{N}}$.*

Proof. Let $(\varphi_{f(e)})_{e \in \mathbb{N}}$ be an enumeration of all primitive recursive functions, with f recursive.

We define the partial recursive function λ by
$$\begin{cases} \lambda(e, 0) = \varphi_e(0) \\ \lambda(e, n+1) = \max(\varphi_e(n+1), \lambda(e, n) + 1). \end{cases}$$

(if one of the two arguments of \max is undefined, \max is undefined)

There is g recursive such that for any $e, n \in \mathbb{N}$, $\lambda(e, n) = \varphi_{g(e)}(n)$.

Hence $(\varphi_{g(f(e))})_{e \in \mathbb{N}}$ is an enumeration of all strictly increasing primitive recursive functions.

We now define the partial recursive function λ' by $\lambda'(e, n) = \text{least } k \varphi_e(k) \geq n$.

Then again there is g' recursive such that for any $e, n \in \mathbb{N}$, $\lambda'(e, n) = \varphi_{g'(e)}(n)$. Setting $l(e) = g'(g(f(e)))$, we obtain that $(\varphi_{l(e)})_{e \in \mathbb{N}}$ is an enumeration of inverses of strictly increasing primitive recursive functions. \square

Fact 4.8. For \mathbf{C} one of our time-complexity classes and $e \in \mathbb{N}$, let $d_{C,e}$ be the martingale defined by $d_{C,e}(x) = d_C((e)_0, x)$, for $x \in \{0, 1\}^*$ (d_C was defined in 4.4), and let h_e be the order $\varphi_{l((e)_1)}$.

Then $(d_{C,e}, h_e)_{e \in \mathbb{N}}$ is an enumeration of all couples (d, h) where d is a martingale in \mathbf{C} such that $d(\emptyset) \leq 1$ and h is the inverse of a strictly increasing primitive recursive function.

Proof. (Of Proposition 4.3) We adapt and simplify Wang's arguments (see [Wan99, Thm 5]): there is no need to encode non-recursive information into the sequence ξ which separates the notions of \mathbf{C} -S randomness and of \mathbf{C} randomness (being an order is a non effective notion whereas - as we saw - being the inverse of a strictly increasing primitive recursive function is an effective one).

We shall thus define by induction two functions F and T both in \mathbf{P} , with $F : \{0, 1\}^* \rightarrow \mathbb{Q}_2$ a martingale and $T : \{0, 1\}^* \rightarrow \mathbb{N}$ monotone (i.e. if $x \preceq y$, then $T(x) \leq T(y)$). Let L be a machine which computes the (total) function l of Lemma 4.7, and let M_u be a universal machine.

- **Level 0.** Let $F(\emptyset) = 1$
 $T(\emptyset) = 0$.

- **Level $s+1$.** We assume $F(x), T(x)$ are defined for $|x| \leq s$ and $T(x) \leq |x|$. As in [Wan99], one distinguishes two cases:

Case 1: For each $e \leq T(x)$, L on input $(e)_1$ stops in $\leq |x| + 1$ steps and there is $m_e \leq |x|$ such that M_u on input $(L((e)_1), m_e)$ stops in $\leq |x| + 1$ steps (outputting $\varphi_{l((e)_1)}(m_e) = h_e(m_e)$), and such that $e + T(x) + 3 < h_e(m_e)$.

Then one sets
$$\begin{cases} F(x0) = 2F(x) \\ F(x1) = 0 \end{cases} \quad \text{and} \quad \begin{cases} T(x0) = T(x) + 1 \\ T(x1) = T(x). \end{cases}$$

Case 2: Otherwise one sets $F(x0) = F(x1) = F(x)$ and $T(x0) = T(x1) = T(x)$.

Let us note that $T(x)$ is simply the number of times case 1 has occurred along x . We also notice that F and T are computable in polynomial time.

The inductive definition of the infinite sequence ξ_C is as follows:

Definition 4.9. Let $s \in \mathbb{N}$. We assume $\xi_C \upharpoonright s$ is defined.

- If $\xi_C \upharpoonright s$ is in case 1, then one sets $\xi_C(s) = 0$.
- Otherwise, one sets $\xi_C(s) = i$ where $i \in \{0, 1\}$ is such that
$$\delta_C((\xi_C \upharpoonright s)i) \leq \delta_C((\xi_C \upharpoonright s)(1-i)).$$

From this definition and Fact 4.6, one deduces:

Fact 4.10.

- If $\mathbf{C} = \mathbf{P}$, then $\xi_C \in \text{FDTIME}(n^{\lceil \log n \rceil})$.
- If $\mathbf{C} = \mathbf{EXP}$, then $\xi_C \in \text{FDTIME}(2^{n^{\lceil \log n \rceil}})$.
- If $\mathbf{C} = \mathbf{TOWER-EXP}$, then $\xi_C \in \text{FDTIME}(T(\lceil \log n \rceil, n))$.
- If \mathbf{C} is **PRIM-REC**, then ξ_C is recursive.

The great lines of the proof are essentially Wang's ones. A trustful reader can skip our proof. However since we simplified the argument (for instance, deleting mention of $F(x)$ in the definition of case 1) and added the machine L , we provide some arguments.

Claim 4.11. Let $\alpha \in \{0, 1\}^{\mathbb{N}}$. Then $\alpha \upharpoonright s$ is in case 1 infinitely often.

Proof. Let $\alpha \in \{0, 1\}^{\mathbb{N}}$ and $s_0 \in \mathbb{N}$ be fixed.

Since L defines a total function, there must exist $s_1 \geq s_0$ such that for all $e \leq T(\alpha \upharpoonright s_0)$, L on input $(e)_1$ stops in $\leq s_1 + 1$ steps.

Also since for $e \in \mathbb{N}$, $h_e (= \varphi_{L((e)_1)})$ is an order, there must exist $s_2 \geq s_1$ such that for every $e \leq T(\alpha \upharpoonright s_0)$, there is $m_e \leq s_2$ such that $h_e(m_e) > e + T(\alpha \upharpoonright s_0) + 3$.

Hence there is $s \geq s_0$ such that $P(s)$ holds where

$$P(s) = \begin{cases} \text{for each } e \leq T(\alpha \upharpoonright s_0), L \text{ on input } (e)_1 \text{ stops in } \leq s + 1 \text{ steps,} \\ \text{there is } m_e \leq s \text{ such that } M_u \text{ on input } (L((e)_1), m_e) \text{ stops in } \leq s + 1 \\ \text{steps outputting } o_e (= h_e(m_e)) \text{ which satisfies } e + T(\alpha \upharpoonright s_0) + 3 < o_e. \end{cases}$$

Let $s_3 = \min\{s \geq s_0 : P(s)\}$. Then by construction $T(\alpha \upharpoonright s_3) = T(\alpha \upharpoonright s_0)$. Hence $\alpha \upharpoonright s_3$ is in case 1. \square

Claim 4.12. $\lim_{s \rightarrow \infty} F(\xi_C \upharpoonright s) = \lim_{s \rightarrow \infty} T(\xi_C \upharpoonright s) = +\infty$. Hence ξ_C is not \mathbf{P} random.

Proof. By definition of F , T and ξ_C , for $s \in \mathbb{N}$,

- $\xi_C \upharpoonright s$ is in case 1, $F(\xi_C \upharpoonright s + 1) = 2F(\xi_C \upharpoonright s)$ and $T(\xi_C \upharpoonright s + 1) = T(\xi_C \upharpoonright s) + 1$,
- when $\xi_C \upharpoonright s$ is in case 2, $F(\xi_C \upharpoonright s + 1) = F(\xi_C \upharpoonright s)$ and $T(\xi_C \upharpoonright s + 1) = T(\xi_C \upharpoonright s)$.

Hence we can conclude by the previous claim. \square

Claim 4.13. For any $s \in \mathbb{N}$, $\delta_C(\xi_C \upharpoonright s) < 2^{T(\xi_C \upharpoonright s) + 2}$.

Proof. (Sketch) Note that $\Phi_C(\emptyset) \leq 2$. Hence $\delta_C(\emptyset) \leq 2 + 2 = 2^2$.

Now δ_C is a martingale, hence for $x \in \{0, 1\}^*$, $i \in \{0, 1\}$, $\delta_C(xi) \leq 2\delta_C(x)$. By clause (b) in definition 4.9, there is an increase of $\delta_C(x)$ (\leq than multiplication by 2) only when case 1 occurs, and case 1 has occurred $T(\xi_C \upharpoonright s)$ times along $\xi \upharpoonright s$. \square

Claim 4.14. For any $e \in \mathbb{N}$, $e + T(\xi_C \upharpoonright s) + 2 < h_e(s)$ a.e. (relatively to s)

Proof. (Sketch)

Let $e \in \mathbb{N}$ be fixed. By Claim 4.12, there is s_0 such that $T(\xi_C \upharpoonright s_0) > e$. By Claim 4.11, there is $s_1 \geq s_0$ such that $\xi_C \upharpoonright s_1$ is in case 1.

One then checks by induction on $s \geq s_1 + 1$ that $e + T(\xi_C \upharpoonright s) + 2 < h_e(s)$.

- If $\xi_C \upharpoonright s$ is in case 1, then there is $m_e \leq s$ such that $e + T(\xi_C \upharpoonright s) + 3 < h_e(m_e) \leq h_e(s)$.

Hence $e + T(\xi_C \upharpoonright s + 1) + 2 = e + T(\xi_C \upharpoonright s) + 3 < h_e(s) \leq h_e(s + 1)$.

- If $\xi_C \upharpoonright s$ is in case 2, this follows directly from the induction hypothesis. \square

One derives from the previous claims:

Claim 4.15. For any $e \in \mathbb{N}$, $d_{C,e}(\xi_C \upharpoonright s) < 2^{h_e(s)}$ a.e.

Proof. For any $e, s \in \mathbb{N}$, $d_{C,e}(\xi_C \upharpoonright s) \leq 2^e \Phi_C(\xi_C \upharpoonright s) \leq 2^e \delta_C(\xi_C \upharpoonright s) \leq 2^e 2^{T(\xi_C \upharpoonright s) + 2}$.

Since $e + T(\xi_C \upharpoonright s) + 2 < h_e(s)$ a.e., we deduce $d_{C,e}(\xi_C \upharpoonright s) < 2^{h_e(s)}$ a.e. \square

One can now conclude: for our classes \mathbf{C} , any true \mathbf{C} order is a true **PRIM-REC** order, hence by Claim 3.23(a) and fact 4.8, given any couple (d, h) such that d is a martingale in \mathbf{C} , with $d(\emptyset) \leq 1$ and h is a true \mathbf{C} order, there is $e \in \mathbb{N}$ such that $d = d_{C,e}$ and $h_e \leq h$. Hence ξ_C is \mathbf{C} -S random. By claim 4.12, ξ_C is not \mathbf{P} random. Finally Fact 4.10 gives the complexity of ξ_C . \square

4.2. Subcomputable weak randomness. We compared the notion of \mathbf{C} -S randomness with the stronger notion of \mathbf{C} randomness. In this subsection, we shall study the relation of \mathbf{C} -S randomness with the weaker notion of “Kurz \mathbf{C} randomness”. There are two candidates for the notion. Wang [Wan00, Definition 5] proposed a notion in terms of martingales and orders:

Definition 4.16 (Wang). Let \mathbf{C} be a class of functions and let $\xi \in \{0, 1\}^{\mathbb{N}}$.

- If d, h are respectively a martingale and an order, then ξ fails the Kurz test (d, h) if $d(\xi \upharpoonright i) \geq h(i)$ a.e.
- ξ is (\mathbf{C}, \mathbf{C}) -W random if ξ passes all Kurz tests for d, h both in \mathbf{C} .

Restricting to the classes **PRIM-REC** or **PSPACE**, Buss, Cenzer and Remmel [BCR14] proposed a different notion (called BP-randomness) and gave three different characterizations in terms of ML-tests, Kolmogorov complexity and martingale property. We give here the martingale and the ML test characterization in the primitive recursive context:

Theorem 4.17 (Buss, Cenzer, Remmel). Let $\xi \in \{0, 1\}^{\mathbb{N}}$.

$$\xi \text{ is BP-random} \quad \text{iff} \quad \begin{cases} \text{for no primitive recursive sequence } (U_n)_{n \in \mathbb{N}} \text{ of clopen sets} \\ \text{such that } \mu(U_n) \leq 2^{-n}, \xi \in \bigcap_{n \in \mathbb{N}} U_n. \\ \text{iff} \quad \begin{cases} \text{for no primitive recursive martingale } d \text{ and for no primitive} \\ \text{recursive function } f \in \mathbb{N}^{\mathbb{N}}, d(\xi \upharpoonright f(n)) \geq 2^n \text{ a.e.} \end{cases} \end{cases}$$

Remark 4.18. In the theorem, one can replace “no primitive recursive function f ” by “no primitive recursive strictly increasing function f ”: in the proof of [BCR14, Thm 2.8 (2) \Rightarrow (1)], the function f can clearly be chosen strictly increasing.

Our notion of \mathbf{C} -S randomness cannot be compared with Wang’s notion of (\mathbf{C}, \mathbf{C}) -W randomness since by [Wan00, Cor 17], for computable sequences, (\mathbf{P}, \mathbf{P}) -W randomness and p-randomness coincide. [BCR14] notion is the right weakening of our notion of \mathbf{C} -S randomness:

Fact 4.19. Let $\xi \in \{0, 1\}^{\mathbb{N}}$.

$$\xi \text{ is BP-random} \quad \Leftrightarrow \quad \begin{cases} \text{for no martingale } d \text{ in } \mathbf{PRIM-REC} \text{ and for no true} \\ \mathbf{PRIM-REC} \text{ order } h, d(\xi \upharpoonright n) \geq 2^{h(n)} \text{ a.e.} \end{cases}$$

Proof.

\Rightarrow : It is possible to derive this result from the ML characterization (see [DH10, 7.2.13] for a similar situation), but it can also be deduced from the above martingale characterization by the method of saving accounts ([DH10, 6.3.8]).

Let $\xi \in \{0, 1\}^{\mathbb{N}}$ and let d, f be primitive recursive such that f is strictly increasing (Remark 4.18) and for $n \geq n_0$, $d(\xi \upharpoonright f(n)) \geq 2^n$.

By considering the function $n \mapsto f(2n+1)$, we can assume for $n \geq n_0$, $d(\xi \upharpoonright f(n)) \geq 2^{2n+1}$. let us define inductively the primitive recursive \mathbb{Q} -valued martingale δ :

- For $|x| \leq f(n_0)$, let $\delta(x) = d(x)$.

– We assume now that $|x| \geq f(n_0)$ and for any $y \preceq x$, $\delta(y)$ is defined.

Let $n_0 \leq n \leq |x|$ be such that $f(n) \leq |x| < f(n+1)$. For $i \in \{0,1\}$, one sets

$$\delta(xi) = \frac{\delta(x \upharpoonright f(n))}{2} + (\delta(x) - \frac{\delta(x \upharpoonright f(n))}{2}) \frac{d(xi)}{d(x)}.$$

This defines a primitive recursive martingale and one checks by induction on $n \geq n_0$, that for any $x \in \{0,1\}^*$, if $|x| = f(n)$, then $\frac{\delta(x)}{d(x)} \geq 2^{-(n-n_0)}$.

By definition of δ , for any x such that $f(n) \leq |x| \leq f(n+1)$, $\delta(x) \geq (1/2)\delta(x \upharpoonright f(n))$.

Hence one deduces for any m, n such that $f(n) \leq m < f(n+1)$,

$$\delta(\xi \upharpoonright m) \geq (1/2)\delta(\xi \upharpoonright f(n)) \geq 2^{-(n-n_0+1)}d(\xi \upharpoonright f(n)) \geq 2^{-(n-n_0+1)}2^{2n+1} \geq 2^n.$$

Let us define $h = \text{Inv}_f \dot{+} 1$. Then h is a true **PRIM-REC** order because f is strictly increasing. If $m > f(n_0)$ and $f(n) \leq m < f(n+1)$, then $0 < \text{Inv}_f(m) \leq n+1$ and hence $h(m) \leq n$. Therefore $\delta(\xi \upharpoonright m) \geq 2^n \geq 2^{h(m)}$, for any $m > f(n_0)$. (Rigorously δ is \mathbb{Q} -valued and not \mathbb{Q}_2 -valued, but we can approximate it in \mathbb{Q}_2 and apply Lemma 3.25).

\Leftarrow : Let us assume $d(\xi \upharpoonright n) \geq 2^{h(n)}$ a.e. for h a true **PRIM-REC** order. Then Inv_h is primitive recursive. For $k \in \mathbb{N}$, if $n_k = \text{Inv}_h(k)$, then $h(n_k) \geq k$. Hence for any $k \in \mathbb{N}$,

$$d(\xi \upharpoonright \text{Inv}_h(k)) = d(\xi \upharpoonright n_k) \geq 2^{h(n_k)} \geq 2^k. \quad \square$$

Hence by the previous fact, **PRIM-REC**-S randomness implies the BP-randomness of [BCR14].

Schnorr [Sch71b] showed that any p-random sequence ξ satisfies the law of Large Numbers (that is if $s_n(\xi) = \sum_{k \leq n} \xi(k)$, then $\lim_n s_n(\xi)/n = 1/2$). As noted in [Wan96, Thm 5.1.8], his argument applies to **(P,P)**-S random sequences. This is also the case for **P**-S random sequences:

Theorem 4.20 (Schnorr).

*Every **P**-S random sequence satisfies the law of Large Numbers.*

Proof. We refer to the exposition of Schnorr's theorem in [Wan96, Thm 5.2.12] and mention here only the small modification at the end. Let us suppose $\xi \in \{0,1\}^{\mathbb{N}}$ does not satisfy the law of Large numbers. One can assume w.l.o.g. $\limsup_n s_n(\xi)/n > 1/2$.

Let $a = (\limsup_n s_n(\xi)/n) - (1/2) > 0$ and let q in $\mathbb{Q}_2 \cap (0,1)$ be so that

$$\frac{1}{2}(\log(1+q) + \log(1-q)) + a(\log(1+q) - \log(1-q)) = c > 0.$$

Defining the \mathbb{Q}_2 -valued martingale F in **P** as in [Wan96] by:

$$\begin{aligned} F(\emptyset) &= 1 \\ F(x1) &= (1+q)F(x) \\ F(x0) &= (1-q)F(x), \end{aligned}$$

one checks that $\limsup_n \log(F(\xi \upharpoonright n))/n = c$. Hence $\log(F(\xi \upharpoonright n))/n \geq c/2$ i.o. Taking $k_0 \in \mathbb{N}$ such that $1/k_0 \leq c/2$ and setting $h(n) = \lfloor n/k_0 \rfloor$, for $n \in \mathbb{N}$, one obtains that $F(\xi \upharpoonright n) \geq 2^{h(n)}$ i.o. Since h is a true **P**-order, ξ cannot be **P**-S random. \square

On the opposite, it is known [kur81] that weak randomness does not imply satisfaction of the law of Large Numbers. In the context of primitive recursiveness, Buss, Cenzer and Remmel obtained the following:

Theorem 4.21 ([BCR14, Thm 2.16]). *There exists a computable BP-random sequence which does not satisfy the law of Large Numbers.*

Hence **PRIM-REC**-S randomness is strictly stronger than BP-randomness (even for computable sequences).

4.3. A summary. In order to summarize all results (some already known, some obtained here) in two tableaux, we agree on the following definitions (only the third one is new):

Definition 4.22. For a class \mathbf{C} and $\xi \in \{0, 1\}^{\mathbb{N}}$ (the martingales are \mathbb{Q}_2 -valued).

- ξ is \mathbf{C} random iff for no martingale d in \mathbf{C} , $\limsup_{n \in \mathbb{N}} d(\xi \upharpoonright n) = +\infty$.
- ξ is \mathbf{C} -S random iff for no martingale d in \mathbf{C} and no true \mathbf{C} order h ,
 $d(\xi \upharpoonright n) \geq 2^{h(n)}$ i.o
- ξ is \mathbf{C} -W random iff for no martingale d in \mathbf{C} and no true \mathbf{C} order h ,
 $d(\xi \upharpoonright n) \geq 2^{h(n)}$ a.e.

If \mathbf{C} is the class of computable functions, then this corresponds to the classical notions of computable randomness, Schnorr randomness and weak randomness.

Remark 4.23. If an infinite binary sequence ξ is in the class \mathbf{C} , then ξ is not \mathbf{C} -W random.

To see this is true, we cannot simply say $\xi \in \bigcap_{n \in \mathbb{N}} [\xi \upharpoonright n]$. The equivalence between the ML definition and the martingale definition has only been shown for the class **PRIM-REC** (Theorem 4.17) and may be problematic for low time-complexity classes.

To justify the remark, let us note that if ξ is in the class \mathbf{C} , then one can consider the martingale d defined as $d(x) = \sum_{i \in \mathbb{N}} 2^i \mu([\xi \upharpoonright 2i] | x)$. d is a \mathbb{Q}_2 -valued martingale in \mathbf{C} and for any $j \in \mathbb{N}$, $d(\xi \upharpoonright j) \geq 2^{\lfloor j/2 \rfloor}$. The function $j \mapsto \lfloor j/2 \rfloor$ is a true \mathbf{P} order, hence ξ is not \mathbf{C} -W random.

In the following tableau, no implication can be reversed:

Computable randomness	\Rightarrow	Schnorr randomness	\Rightarrow	weak randomness
(1) \Downarrow		(1) \Downarrow		(1) \Downarrow
PRIM-REC randomness	\Rightarrow	PRIM-REC -S randomness	\Rightarrow	PRIM-REC -W randomness
	(2)		(3)	

The impossibility of reversing the implications in the first line is classical (Schnorr, Wang). Concerning the other implications:

- (1) let $\mathbf{C} \subsetneq \mathbf{C}'$ be two classes in our collection of time-complexity classes, or let \mathbf{C} be in our collection and let \mathbf{C}' be the class of recursive functions. It is known that there is a sequence ξ in \mathbf{C}' which is \mathbf{C} random (One can use the martingale $\Psi_{\mathbf{C}}(x) = \sum_e 2^{-e} d_{\mathbf{C}}(e, x)$, with $d_{\mathbf{C}}$ given in Definition 4.4). By Remark 4.23, ξ is not \mathbf{C}' -W random,
- (2) is by Proposition 4.3(i),
- (3) is by Proposition 4.20 and Theorem 4.21.

In the next tableau, the non-reversibility of the implication holds also when the notion is restricted to the class of computable sequences. (1), (2) and (3) refer to the above justifications (we abbreviate **TOWER-EXP** to **TO-EXP** to be able to insert the tableau).

PRIM-REC randomness	\Rightarrow (2)	PRIM-REC-S randomness	\Rightarrow (3)	PRIM-REC-W randomness
(1) \Downarrow		(1) \Downarrow		(1) \Downarrow
TO-EXP randomness	\Rightarrow (2)	TO-EXP-S randomness	\Rightarrow (3)	TO-EXP-W randomness.
(1) \Downarrow		(1) \Downarrow		(1) \Downarrow
EXP randomness	\Rightarrow (2)	EXP-S randomness	\Rightarrow (3)	EXP-W randomness
(1) \Downarrow		(1) \Downarrow		(1) \Downarrow
P randomness	\Rightarrow (2)	P-S randomness	\Rightarrow (3)	P-W randomness

REFERENCES

- [AM97] Klaus Ambos-Spies and Elvira Mayordomo. Resource-bounded measure and randomness. In *Complexity, logic, and recursion theory*, pages 1-47. New York, NY: Marcel Dekker, 1997.
- [ATZ94] Klaus Ambos-Spies, Sebastiaan A. Terwijn and Xizhong Zheng. Resource bounded randomness and weakly complete problems. In *Algorithms and computation. 5th international symposium, ISAAC '94, Beijing, P. R. China, August 25-27, 1994. Proceedings*, pages 369-377. Berlin: Springer.
- [BCR14] S. Buss, D. Cenzer and J. Remmel. Sub-computable bounded randomness. *Log. Methods Comput. Sci.*, vol. 10 issue 4, Article 15, 2014.
- [Cha75] G. Chaitin. A theory of program size formally identical to information theory. *Journal of ACM*, vol. 22, pages 329-340, 1975.
- [CSZ07] Qingliang Chen, Kaile Su and Xizhong Zheng. Primitive recursive real numbers. *Math. Log. Quart.*, Vol. 53, (4-5), pages 365-380, 2007.
- [DG04] R.G Downey and E. Griffiths. Schnorr randomness. *Jour. Symb. Log.* Vol. 69(2) pages 533-554, 2004.
- [DH10] R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer-Verlag, 2010.
- [JL95] David W. Juedes and Jack H. Lutz. Weak completeness in E and E₂. *Theor. Comput. Sci.*, Vol.143(1), Pages 149-158. Elsevier, Amsterdam, 1995.
- [Ko86] K. Ko. On the notion of infinite pseudorandom sequences. *Theoret. Comput. Sci.*, vol.48, pages 9-33, 1986.
- [kur81] S.Kurz. *Randomness and Genericity in the degrees of unsolvability*. Ph.D. thesis, University of illinois, 1981.
- [Lev74] L. A. Levin. The concept of a random sequence. *Soviet Math. Dokl.*, vol. 14, pages 1413-1416, 1974.
- [Lut90] J.H. Lutz. Category and measure in complexity classes. *SIAM J. Comput.*, Vol. 19(6), pages 1100-1131, 1990.
- [Lut92] J.H. Lutz. Almost everywhere high nonuniform complexity. *Journal Comput. System Sci.*, vol. 44, pages 220-258, 1992.
- [Mar66] Per Martin-Löf. The definition of random sequences. *Inf. Control*, Vol. 9, pages 602-619 Academic Press, New York etc., 1966.
- [May94] E. Mayordomo. *Contributions to the study of Resource-bounded Measure*, Ph.D. thesis, Barcelona University, 1994.
- [Nie09] A. Nies. *Computability and Randomness*. Oxford University Press, 2009.
- [Odi92] P.G. Odifreddi. *Classical recursion theory. The theory of functions and sets of natural numbers*. North-Holland, 1992.
- [Odi99] P.G. Odifreddi. *Classical recursion theory, vol. II*. Elsevier, 1999.
- [Sch71a] C.P. Schnorr. A unified approach to the definition of random sequences. *Math. System Theory*, vol. 5, pages 246-258, 1971.
- [Sch71b] C.P. Schnorr. Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie. (Randomness and probability. An algorithmic foundation of probability theory). Lecture Notes in Mathematics 218. Berlin-Heidelberg-New York: Springer-Verlag, 1971.
- [Sch73] C.P. Schnorr. Process Complexity and effective random tests. *Journal of Computer and System Sciences*, vol. 7, pages 376-388, 1973.
- [Wan96] Y. Wang. Randomness and Complexity, Ph.D. Universität Heidelberg, 1996.

- [Wan99] Y. Wang. A separation of two randomness concepts. *Information Processing Letters*, vol. 69, pages 115-118, 1999.
- [Wan00] Y. Wang. Resource bounded randomness and computational complexity *Theoret. Comput. Sci*, vol. 237, pages 33-55, 2000.