Axiomatizing Resource Bounds for Measure *

Xiaoyang Gu[1] **, Jack H. Lutz[2] ***, Satyadev Nandakumar[3], and James S. Royer[4] †

[1] LinkedIn Corporation,
Mountain View, CA 94043, U. S. A.
xgu@linkedin.com
[2] Iowa State University
Ames, IA 50011, U. S. A.
lutz@cs.iastate.edu
[3] I. I. T. Kanpur
Kanpur, UP 280 016 India.
satyadev@cse.iitk.ac.in
[4] Syracuse University
Syracuse, New York 13244, U. S. A.
royer@ecs.syr.edu

Abstract. Resource-bounded measure is a generalization of classical Lebesgue measure that is useful in computational complexity. The central parameter of resource-bounded measure is the *resource bound* Δ, which is a class of functions. When Δ is unrestricted, i.e., contains all functions with the specified domains and codomains, resource-bounded measure coincides with classical Lebesgue measure. On the other hand, when Δ contains functions satisfying some complexity constraint, resource-bounded measure imposes internal measure structure on a corresponding complexity class.

Most applications of resource-bounded measure use only the "measure-zero/measure-one fragment" of the theory. For this fragment, Δ can be taken to be a class of type-one functions (e.g., from strings to rationals). However, in the full theory of resource-bounded measurability and measure, the resource bound Δ also contains type-two functionals. To date, both the full theory and its zero-one fragment have been developed in terms of a list of example resource bounds chosen for their apparent utility.

This paper replaces this list-of-examples approach with a careful investigation of the conditions that suffice for a class Δ to be a resource bound. Our main theorem says that every class Δ that has the closure properties of Mehlhorn's basic feasible functionals is a resource bound for measure.

We also prove that the type-2 versions of the time and space hierarchies that have been extensively used in resource-bounded measure have these closure properties. In the course of doing this, we prove theorems establishing that these time and space resource bounds are all robust.

Keywords: basic feasible functionals, computational complexity, resource-bounded measure, type-two functionals

1 Introduction

Resource-bounded measure is a generalization of classical Lebesgue measure theory that allows us to quantify the "sizes" (measures) of interesting subsets of various complexity classes. This quantitative capability has been useful in computational complexity because it has intersected informatively with reducibilities, completeness, randomization, circuit-size, and many other central ideas of complexity theory. Resource-bounded

* A preliminary version of the paper was presented in the Workshop on Logic in Computational Complexity, 2009.
** This author's research was supported in part by National Science Foundation Grants 0344187, 0652569 and 0728806.
*** This author's research was supported in part by Spanish Government MEC Project TIN 2005-08832-C03-02.
† This author's research was supported in part by National Science Foundation Grant CCR-0098198.

measure has given us a generalization of the probabilistic method that works inside complexity classes (leading, for example, to improved lower bounds on Boolean circuit size [14] and the densities of complete problems [16]) and new complexity-theoretic hypotheses (e.g., the hypothesis that NP is a non-measure 0 subset of exponential time) with many plausible consequences, i.e., significant explanatory power. The somewhat outdated survey papers [4,2,2,5,17,22] and more recent papers in the bibliography [10] give a more detailed account of the scope of resource-bounded measure and its applications.

The central parameter in resource-bounded measure is the *resource bound*, which is a class Δ of functions. When Δ is unrestricted, i.e., contains *all* functions with specified domains and codomains, resource-bounded measure coincides with classical Lebesgue measure on the Cantor space $\mathbf{C}$ of all decision problems. On the other hand, when Δ only contains functions satisfying a suitable complexity constraint, resource-bounded measure consists of the following two theories.

1. A theory of Δ-*measure*. This is a "Δ-constructive" measure theory on $\mathbf{C}$.
2. A theory of *measure in* a complexity class $R(\Delta)$. This is a theory that Δ-measure imposes on the "result class" $R(\Delta)$.

(Result classes and other notions discussed informally in this introduction are defined precisely in the sections that follow.) For example, if $\Delta = \mathrm{p}$ consists of functions that are computable in polynomial time, then we have p-measure on $\mathbf{C}$, and this imposes an internal measure structure on the exponential time complexity class $R(\mathrm{p}) = \mathrm{E} = \mathrm{DTIME}(2^{\mathrm{linear}})$. Typically, one proves a result on measure in $R(\Delta)$ by proving a corresponding result on Δ-measure. This, together with the fact that the Δ-measure result implies a corresponding Δ'-measure result for every resource bound $\Delta' \supseteq \Delta$, provides resource-bounded measure a substantial underlying unity.

Of the hundred or so papers that have been written about resource-bounded measure since 1992, none gives a definition of the term "resource bound". Most simply work with those few resource bounds appropriate to the complexity-theoretic problems being investigated. Even papers of a more general nature stipulate that the resource bound Δ is one of a specified (infinite) list of examples chosen for their prior utility.

This approach to resource bounds has been healthy for the initial development of a theory intended as a tool, but, as Socrates taught us in *Euthyphro* [23], a list of examples leaves us far short of understanding a concept. More pragmatically, as the list grows, it becomes ever more burdensome to verify that a theorem about a general resource bound Δ actually holds for all examples in the list.

This paper shows that there is a simple and natural set of axioms with the following two properties.

– Adequacy: Any class Δ satisfying the axioms can be used as a resource bound for measure.
– Generality: The most extensively used resource bounds satisfy the axioms.

We thus propose to *define* a resource bound to be a class Δ satisfying the axioms.

What makes our task challenging is the fact that, in order to define resource-bounded measurability and measure [15] a resource bound Δ must contain not only functions on discrete domains like $\{0,1\}^*$ and $\mathbb{N}$, but also type-2 *functionals* that take functions as arguments. It has been a major undertaking to define what it means for such functionals to be feasible (computable in polynomial time) and to verify that the definition is robust [12,8]. The second author [15] has defined type-2 versions of the other time and space resource bounds that have been extensively used in resource-bounded measure (the quasi-polynomial time and space hierarchies). However, these definitions have not been proven to be robust, and the machine-based definitions of [15], while proven to be sufficient for the development of measure and measurability, shed very little light on our present question, namely, what properties of a class of type-2 functionals make it an adequate resource bound for measure.

Fortunately, it turns out that an *existing* set of axioms can be adapted to our purpose. Mehlhorn's *basic feasible functionals* [19] were originally defined as a function algebra, i.e., a set of initial functionals and a set of closure properties, with the understanding that the class of basic feasible functionals is the *smallest* class containing these initial functions and enjoying these closure properties.

The main contribution of the present paper is to demonstrate that, if we just discard the "smallest" proviso in Mehlhorn's scheme and define a resource bound to be *any* class of functionals containing the initial functions and having the closure properties of his definition, then we will, indeed have a definition that is sufficient for the development of measurability and measure in [15].

We also prove that all the classes in the quasi-polynomial time and space hierarchies of [15] are resource bounds in this sense. In the course of proving this, we prove new function algebra characterizations of these classes, thereby establishing that they are robust.

Two additional remarks on related work are in order here. First, there has been work on resource-bounded measure that is not captured by our axiomatization. The notable examples here are the measures in "small" complexity classes (e.g., the polynomial time class P) developed by Moser [20] (building on pioneering work of Mayordomo [18] and Allender and Strauss [1]), the measures in probabilistic classes (e.g., the randomized exponential time class BPE) developed by Moser [21], and the measures in "large" complexity classes (e.g., the doubly exponential time class EE) developed by Harkins and Hitchcock [9]. To date, this work has all been confined to measure 0/measure 1 results. Future developments of general measurability and measure in these settings may necessitate – and guide – generalizations of the axiomatization presented here. This remains an open question.

The other line of related work that we mention is Dai's outer measure approach to measurability and measure in complexity classes [7]. This approach is simpler than that of [15] and the present paper in that it does not require type-two functionals. On the other hand, the approach of [7] only seems to yield theory 2 in the second paragraph of this introduction, so that all results are "local" to a particular complexity class. The unity provided by theory 1 above, i.e., a "global" Δ-measure on all of Cantor space, is a substantial advantage of our our present approach. Only future research will determine whether a single approach can achieve both the simplicity of [7] and the unity of [15].

The rest of the paper is organized as follows. Section 2 gives the preliminary definitions and notational conventions. Section 3 describes the classes of type-two functionals used in the paper. Section 4 gives the definition of a resource bound and shows that the standard resource bounds in the literature satisfy the definition. Section 5 establishes that the definition of a resource bound is adequate to establish the fundamental theorems of resource-bounded measure. The final section proves that the measure-zero fragment of this theory coincides with the approach current in the literature [15].

2 Preliminaries

We use a binary alphabet $\{0,1\}$ in this paper. A string is an element in $\{0,1\}^*$. For every $w \in \{0,1\}^*$, $|w|$ is the length of the string w, and $w[i]$ denotes the ith bit of w. The *Cantor* space $\mathbf{C} = \{0,1\}^\infty$ is the set of all infinite binary sequences. For an $S \in \mathbf{C}$, $S[i]$ is the ith bit of S, and $S[0..n-1]$ is the n-bit prefix of S. The standard enumeration of $\{0,1\}^*$ is the enumeration of all strings in $\{0,1\}^*$ in increasing order of length, with strings of the same length ordered lexicographically. The binary encoding function is $\mathrm{ntob} : \mathbb{N} \to \{0,1\}^*$ such that for all $n \in \mathbb{N}$, $\mathrm{ntob}(n)$ is the nth string in the standard enumeration. The binary decoding function $\mathrm{bton} : \{0,1\}^* \to \mathbb{N}$ is the inverse of the binary encoding function. For example, $\mathrm{bton}(\lambda) = 0$ and $\mathrm{bton}(01) = 4$.

The binary notational successor functions are $s_0, s_1 : \{0,1\}^* \to \{0,1\}^*$ such that $s_0(u) = u0$ and $s_1(u) = u1$ for all $u \in \{0,1\}^*$. The binary successor function is $s : \{0,1\}^* \to \{0,1\}^*$ such that for all $u \in \{0,1\}^*$, $s(u) = \mathrm{ntob}(\mathrm{bton}(u)+1)$) - that is, if u represents a number n, then $s(u)$ is the encoding of $n+1$. The binary predecessor function is $\mathrm{pred} : \{0,1\}^* \to \{0,1\}^*$ such that $\mathrm{pred}(u) = \mathrm{ntob}(\max\{\mathrm{bton}(u) - 1, 0\})$.

The smash function is $\# : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that for all $u, v \in \{0,1\}^*$, $\#(u,v) = 1^{|u|\cdot|v|}$. The interesting property of the smash function is that for every pair (u,v), the string $\#(u,v)$ has length equal to the product of the lengths of u and v.

A language is a subset of $\{0,1\}^*$. The characteristic sequence of L the infinite binary sequence such that $S[i] = 1 \iff \mathrm{ntob}(i) \in L$. Analogously, the characteristic function of L is $\chi_L : \{0,1\}^* \to \{0,1\}$ such that $\chi_L(x) = 1 \iff x \in L$. When no ambiguity arises, we also use L for the characteristic sequence of L.

We write $w \sqsubseteq A$ if string w is a prefix of a string/sequence A. A cylinder in $\mathbf{C}$ is a subset, of the form $\{S \in \mathbf{C} \mid w \sqsubseteq S\}$ for some w, denoted $\mathbf{C}_w$. An *open set* in $\mathbf{C}$ is a set of the form $\bigcup_{w \in A} \mathbf{C}_w$ for some $A \subseteq \{0,1\}^*$.

We also define the following hierarchy of functions. Let $g_0 = 2n$ and let $g_i(n) = 2^{g_{i-1}(\log n)}$ for all $i \in \mathbb{Z}^+$. Note that $g_1(n) = n^2$ and that $g_2(n) = n^{\log n}$. For $i \in \mathbb{N}$, let G_i be the class of functions that contains g_i and is closed under composition. We use G_i to represent different growth rates. G_1 represents polynomial growth rates ($O(n^c)$) and G_2 represents quasi-polynomial growth rates ($O(n^{\log^c n})$). For each $i \in \mathbb{N}$, we call growth rates bounded by a function in G_i as quasi^i-polynomials.

3 Type–2 Functionals

In 1965, Cobham characterized type-1 polynomial-time computable functions using limited/bounded recursion on notation [6,25]. He proved that the class of polynomial-time computable functions is the smallest class of functions containing the constant 0 function, the binary notational successor functions, and the smash function that is closed under composition and limited recursion on notation.

Mehlhorn extended the characterization of polynomial-time computability to type-2 functionals.

Definition 1 (Mehlhorn [19]). *F is defined from G, H, K by* limited recursion on notation *if for all $\boldsymbol{f}$, $\boldsymbol{x}$, w,*

$$\begin{aligned}&F(\boldsymbol{f},\boldsymbol{x},\lambda) = G(\boldsymbol{f},\boldsymbol{x})\\&F(\boldsymbol{f},\boldsymbol{x},wb) = H(\boldsymbol{f},\boldsymbol{x},wb,F(\boldsymbol{f},\boldsymbol{x},w)),\, b \in \{0,1\}\\&|F(\boldsymbol{f},\boldsymbol{x},w)| \le |K(\boldsymbol{f},\boldsymbol{x},w)|.\end{aligned}$$

We also use the following definition from Kapron and Cook [13].

Definition 2 (Kapron and Cook [13]). *F is defined from H, G_1, ..., G_l by* functional composition *if for all $\boldsymbol{f}$, $\boldsymbol{x}$,*

$$F(\boldsymbol{f},\boldsymbol{x}) = H(\boldsymbol{f},G_1(\boldsymbol{f},\boldsymbol{x}),\ldots,G_l(\boldsymbol{f},\boldsymbol{x})).$$

F is defined from G by expansion *if for all $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{x}$, $\boldsymbol{y}$,*

$$F(\boldsymbol{f},\boldsymbol{g},\boldsymbol{x},\boldsymbol{y}) = G(\boldsymbol{f},\boldsymbol{x}).$$

For the definition of basic feasible functionals, we adopt Kapron and Cook's definition.

Definition 3 (Kapron and Cook [13]). *Let X be a set of type-two functionals. The class of* basic feasible functionals defined from X *(*$\mathrm{BFF}(X)$*) is the smallest class of functionals that contains X, all polynomial-time functions of type-one and the application functional* $\mathbf{Ap}$*, defined by $\mathbf{Ap}(f,x) = f(x)$, and is closed under functional composition, expansion, and limited recursion on notation. The* basic feasible functionals *are* $\mathrm{BFF}(\varnothing)$.

Remark 1. In this definition, it is possible to replace the inclusion of polynomial-time functions in $\mathcal{T}_1$ with the inclusion of the constant 0 function, the binary notational successor functions, and the smash function. But since Cobham's functional algebraic characterization of polynomial-time is well-understood now, directly using polynomial-time functions allow us to avoid repeating the tedious process of defining all the simple functions from scratch.

Mehlhorn proved that the BFF's have the *Ritchie-Cobham property*, namely, $F \in \text{BFF}$ if and only if there exists an oracle Turing machine M and $G \in \text{BFF}$ such that for all input f and x, the running time of $M(f, x)$ is bounded by $|G(f, x)|$. Mehlhorn's result serves as partial evidence that the functional algebraic notion of BFF is robust. On top of this, Kapron and Cook defined a notion of type-2 polynomial-time computability based on oracle Turing machines that does not require the use of BFF time-bound like the one in Mehlhorn's result. The Basic Feasible Functionals capture the notion of an intuitively feasible class of type-two functionals. Basic Feasible Functionals and their probabilistic versions show up in cryptography, for instance, in many constructions of pseudo-random generators from one way functions. In [11] the authors remark that many cryptographic adversaries can be formalized as type-2 probabilistic feasible functionals or circuits.

First, we generalize Kapron and Cook's definition of second-order polynomials to the following.

Definition 4. *Let $i \in \mathbb{Z}^+$.* First-order variables *are elements of the set $\{n_1, n_2, \dots\}$.* Second-order variables *are elements of the set $\{L_1, L_2, \dots\}$.* Second-order quasi^i-polynomials *are defined inductively: any $c \in \mathbb{N}$ is a second-order quasi^i-polynomial; first-order variables are second-order quasi^i-polynomials; and if P, Q are second-order quasi^i-polynomials and L is a second-order variable, then $P + Q$, $P \cdot Q$, $L(P)$, and $g_i(P)$ are second-order quasi^i-polynomials.*

Second-order quasi^1-polynomials are the second-order polynomials defined by Kapron and Cook. Second-order quasi^2-polynomials are second-order quasi-polynomials. They also defined a notion of the length for type-1 functions.

Definition 5 (Kapron and Cook[13]). *For any $f : \{0,1\}^* \to \{0,1\}^*$, the* length *of f is the function $|f| : \mathbb{N} \to \mathbb{N}$ defined by*

$$|f|(n) = \max_{|w| \le n} |f(w)|.$$

Note that $|f|$ is non-decreasing.

With the above two definitions, Kapron and Cook defined the following notion of polynomial-time bounded oracle Turing machine computation.

Definition 6. *A type-two functional F is basic poly time if there is an oracle Turing machine M and a second-order polynomial P such that M computes F, and for all $\boldsymbol{f}$ and $\boldsymbol{x}$, the running time of $M(\boldsymbol{f}, \boldsymbol{x})$ is bounded by $P(|f_1|, \dots, |f_k|, |x_1|, \dots, |x_l|)$.*

Strongly confirming the robustness of the notion of BFFs, they proved the following.

Theorem 1 (Kapron and Cook [13]). *A functional F is* BFF *if and only if it is basic poly time.*

In this paper, we extend the Mehlhorn's functional algebraic notion of feasible functionals to quasi-feasible functionals with the following definition.

Definition 7. *Let $X \subseteq \mathcal{T}_2$ and let $i \in \mathbb{Z}^+$. The class of* basic i-feasible functionals defined from X *($\text{BFF}_i(X)$) is the smallest class of functionals containing X, all polynomial-time computable functions in $\mathcal{T}_1$, $1^{g_i(|x|)}$,* *and the application functional* $\mathbf{Ap}$*, defined by* $\mathbf{Ap}(f, x) = f(x)$*, and which is closed under functional composition, expansion, and limited recursion on notation. The* basic i-feasible functionals *are elements of the class* $\mathrm{BFF}_i(\varnothing)$.

In the flavor of Kapron and Cook, we extend their oracle Turing machine based notion of feasible computation to the following.

Definition 8. *Let* $i \in \mathbb{Z}^+$*. A functional* F *is basic quasi*^{i}*-polynomial time if there is an oracle Turing machine* M *and a second-order quasi*^{i}*-polynomial* P *such that* M *computes* F*, and for all* $\boldsymbol{f}$ *and* $\boldsymbol{x}$*, the running time of* $M(\boldsymbol{f}, \boldsymbol{x})$ *is bounded by* $P(|f_1|, \ldots, |f_k|, |x_1|, \ldots, |x_l|)$.

The following theorem is a corollary of Kapron and Cook's proof of theorem 1.

Theorem 2. *Let* $i \in \mathbb{Z}^+$*. A functional* F *is* BFF_i *if and only if it is basic quasi*^{i}*-polynomial time.*

In the machine model, the time bound is based on both the input length and on the length of query answers. This is why we need to have $g_i(P)$ in the definition of second-order quasi^i-polynomials. The condition in the definition that a single second-order quasi^i-polynomial has to work for all input $\boldsymbol{f}$ prohibits an oracle Turing machine from using extra running time when the input function $\boldsymbol{f}$ is pathologically long. An oracle Turing machine M that computes a quasi^i-polynomial time functional, on any x, can only utilize an amount of time that is quasi^i-polynomial in the length of $\boldsymbol{f}$ it can provide evidence for, which can be much less than length of $\boldsymbol{f}$ depending on the type-0 inputs.

More formally, let Q_x be the set of all queries made by M with $\boldsymbol{f}$ and x as input. Let P be the time bound of M. Let $f_{Q_x}(y) = \boldsymbol{f}(y)$ if $y \in Q_x$ and 0 otherwise. Then the running time $T_M(\boldsymbol{f}, x) \leq P(|f_{Q_x}|, |x|)$ for all $\boldsymbol{f}$ and x. The key idea behind Kapron and Cook's proof is that it is possible to find the oracle query $q_{\max}$ made by $M(\boldsymbol{f}, x)$ that maximizes $|f(q_{\max})|$ in BFF. And the inability to compute the length of $\boldsymbol{f}$ (in unary) in BFF is what makes their proof very involved. We will see in the following that the situation with polynomial space-bounded computation is much simpler precisely for the reason that, as we will soon prove in Lemma 1, the length functional in unary for arbitrary $\boldsymbol{f}$ is actually computable in polynomial space. First, we develop the definitions of computation feasible in terms of space.

Definition 9. *A functional* F *is* quasi^i-polynomial space *if there is an oracle Turing machine* M *and a second order quasi*^{i}*-polynomial* P *such that* M *computes* F*, and for all* $\boldsymbol{f}$*,* $\boldsymbol{x}$*,* $S_M(\boldsymbol{f}, \boldsymbol{x})$ *is bounded by* $P(|f_1|, \ldots, |f_k|, |x_1|, \ldots, |x_l|)$*, where* $S_M(\boldsymbol{f}, \boldsymbol{x})$ *is the running space used by* M *on input* $\boldsymbol{f}$ *and* $\boldsymbol{x}$*.*

In 1972, D. B. Thompson characterized the class of type-1 polynomial-space computable functions as the smallest class that contains the constant 0 function, the binary successor function, the smash function, and is closed under (type-1) composition and (type-1) bounded recursion [24]. We extend type-1 bounded recursion as follows.

Definition 10. F *is defined from* G*,* H*,* K *by* bounded recursion *(BR) if for all* $\boldsymbol{f}$*,* $\boldsymbol{x}$*,* n*,*

$$\begin{aligned} &F(\boldsymbol{f}, \boldsymbol{x}, 0) = G(\boldsymbol{f}, \boldsymbol{x}) \\ &F(\boldsymbol{f}, \boldsymbol{x}, n+1) = H(\boldsymbol{f}, \boldsymbol{x}, n, F(\boldsymbol{f}, \boldsymbol{x}, n)) \\ &F(\boldsymbol{f}, \boldsymbol{x}, n) \leq K(\boldsymbol{f}, \boldsymbol{x}, n). \end{aligned}$$

Definition 11. *Let* $X \subseteq \mathcal{T}_2$ *and let* $i \in \mathbb{Z}^+$*. The class of* basic i-feasible space functionals defined from X *(*$\mathrm{BFSF}_i(X)$*) is the smallest class of functionals containing* X*, all polynomial-time computable functions in* $\mathcal{T}_1$*,* $1^{g_i(|x|)}$*, and the application functional* $\mathbf{Ap}$*, defined by* $\mathbf{Ap}(f, x) = f(x)$*, and which is closed under functional composition, expansion, and bounded recursion. The* basic i-feasible space functionals *are* $\mathrm{BFSF}_i(\varnothing)$.

Lemma 1. *$L : (f, x) \mapsto 1^{|f|(|x|)}$ is basic i-feasible space for all $i \geq 1$.*

Proof (Proof of Lemma 1). Let the functional

$$F : (\{0,1\}^* \to \{0,1\}^*) \times \{0,1\}^* \to \{0,1\}^*$$

be defined using bounded recursion as follows:

$$F(f, \lambda) = \lambda$$
$$F(f, x) = \begin{cases} F(f, \mathrm{pred}(x)) & |f(F(f, \mathrm{pred}(x)))| \geq |f(x)| \\ x & \text{otherwise} \end{cases}$$
$$F(f, x) \leq x.$$

Intuitively,

$$F(f, x) = \min \left\{ y \;\middle|\; y \leq x \text{ and } |f(y)| = \max_{z \leq x} |f(z)| \right\}.$$

Let

$$L(f, x) = 1^{|\mathbf{Ap}(f(F(f, 1^{|x|})))|}.$$

Then L is the functional we desire here. □

Theorem 3. *A functional F is basic i-feasible space if and only if it is quasi^i-polynomial space.*

Proof (Proof Sketch of Theorem 3.). To prove that basic i-space feasibility implies quasi^i-polynomial space, it suffices to do an induction on the structure of composition and bounded recursion by implementing them on Turing machine with space reuse.

For the other side of the equivalence, we prove this by an induction on the depth of second-order polynomials.

Let F be a quasi^i-polynomial space computable functional computed by OTM M with space bound P of depth $d > 0$. (When $d = 0$, the running space bound of F does not depend on the length of the input type-1 function f and the proof is simpler.) Then there exist (regular) level-i polynomials $q_0, q_1, \ldots, q_d$, and second-order quasi^i-polynomials $P_1, \ldots, P_d$ such that for all $i \in [1..d-1]$

$$P_i(|f|, n) = |f|(q_{i-1}(n))$$

and

$$P_d(f, x) = q_d(P_{d-1}(|f|, n)) \geq P(|f|, n).$$

The following pseudo-code provides a functional $B(f, x)$ that computes the space bound of the OTM M with input f and x. The description of $B(f, x)$ is written in imperative programming language style pseudo-code. Note that d is a fixed constant, it is easy to transform this pseudo-code into functional algebra simply using d levels of composition of the functional L.

---

input f, x
$q_{\max} = \lambda$
for $i = 0$ **to** $d - 1$
    $q_{\max} = L(f, 1^{q_i(|x|)})$
return $1^{|\mathbf{Ap}(f, q_{\max})|}$

As soon as we have the actual space bound $B(f,x)$ of the computation of M on input f and x and hence the bound of number of (transition) steps $M^f(x)$ takes to run, we can define a functional Run_M similar to Kapron and Cook. Our functional Run_M differs from theirs mainly in two aspects. One is that ours keeps track of only the encoding of the instantaneous description of the Turing machine at the current computation step, while theirs keeps track of the encoding of the entire history of the computation of the Turing machine. The other is that our Run_M uses bounded recursion, while theirs uses bounded recursion on notation. The techniques used in transforming Turing machine transition function to functional algebra are standard, though tedious. $Run_M(f,x,y)$ recurses on the value of y and $Run_M(f,x,B(f,x))$ is the instantaneous description at the time $M(f,x)$ halts and

$$Run_M(f,x,B(f,x)) \leq B(,x)$$

for all f and x. □

4 Resource Bounds

In the initial development of a theory of resource-bounded measure [15], a list of examples of resource-bounds were given based on an oracle Turing machine model of type-2 computation that is not known to be robust. In this section, we axiomatize the definition of a resource bound by adapting the axioms of Mehlhorn's basic feasible functionals and verify that most extensively used resource bounds are indeed resource bounds under this definition.

Definition 12. *A* resource bound *is a class Δ of functionals of type no more than* 2 *that is closed under* BFF*.*

Theorem 4. *Let $i \in \mathbb{Z}^+$.* $\mathrm{p}_i = \mathrm{BFF}_i$ *is a resource bound.*

Proof (Proof of Theorem 4). Note that by definition $\mathrm{BFF}_i = \mathrm{BFF}(\{x \mapsto 1^{g_i(|x|)}\})$. Since $\mathrm{BFF}(\mathrm{BFF}(X)) = \mathrm{BFF}(X)$ for all X, $\mathrm{BFF}(\mathrm{BFF}_i) = \mathrm{BFF}_i$. Therefore BFF_i is a resource bound. □

Let K^k be the canonical Σ_k^{P}-complete language [3]. Let χ_k be the characteristic function of K^k.

Definition 13. *Let $i \in \mathbb{Z}^+$ and let $k \geq 2$.* $\Delta_k^{\mathrm{p}_i} = \mathrm{BFF}_i(\{\chi_{k-1}\})$*.*

Theorem 5. *Let $i \in \mathbb{Z}^+$ and let $k \geq 2$. $\Delta_k^{\mathrm{p}_i}$ is a resource bound.*

Proof (Proof of Theorem 5). Note that by definition $\Delta_k^{\mathrm{p}_i} = \mathrm{BFF}(\{x \mapsto 1^{g_i(|x|)}, \chi_{k-1}\})$. Since $\mathrm{BFF}(\mathrm{BFF}(X)) = \mathrm{BFF}(X)$ for all X, $\mathrm{BFF}(\Delta_k^{\mathrm{p}_i}) = \mathrm{BFF}(\mathrm{BFF}(\{x \mapsto 1^{g_i(|x|)}, \chi_{k-1}\})) = \mathrm{BFF}(\{x \mapsto 1^{g_i(|x|)}, \chi_{k-1}\}) = \Delta_k^{\mathrm{p}_i}$. Therefore $\Delta_k^{\mathrm{p}_i}$ is a resource bound. □

Definition 14. *Let $i \in \mathbb{Z}^+$.* $\mathrm{p}_i\mathrm{space} = \mathrm{BFSF}_i$*.*

Theorem 6. *Let $i \in \mathbb{Z}^+$.* $\mathrm{BFF}(\mathrm{p}_i\mathrm{space}) = \mathrm{p}_i\mathrm{space}$*, i.e.,* $\mathrm{p}_i\mathrm{space}$ *is a resource bound.*

Proof (Proof of Theorem 6.). We prove the equivalence for $i = 2$. Since each of the G_i is closed under composition, the proof readily extends to all $i \in \mathbb{Z}^+$.

It suffices to show that $\mathrm{BFF}(\mathrm{p}_2\mathrm{space}) \subseteq \mathrm{p}_2\mathrm{space}$, i.e., $\mathrm{p}_2\mathrm{space}$ is closed under functional composition, expansion, and limited recursion on notation.

Functional composition

Let $H, G_1, \ldots, G_l \in \mathrm{p}_2\mathrm{space}$. Let F be defined from $H, G_1, \ldots, G_l$ by functional composition, i.e., for all $\boldsymbol{f} \in \mathrm{p}_2\mathrm{space}$ and $\boldsymbol{x} \in \{0,1\}^*$,

$$F(\boldsymbol{f}, \boldsymbol{x}) = H(\boldsymbol{f}, G_1(\boldsymbol{f}, \boldsymbol{x}), \ldots, G_l(\boldsymbol{f}, \boldsymbol{x})).$$

Now, we show that $F \in \mathrm{p}_2\mathrm{space}$.

Since $H, G_1, \ldots, G_l \in \mathrm{p}_2\mathrm{space} \cap \mathcal{T}_2$, there exist oracle Turing machines M_H, $M_1, \ldots, M_l$ and second-order polynomial space bounds $P_H, P_1, \ldots, P_l : ((\mathbb{N} \to \mathbb{N}) \times \mathbb{N}) \to \mathbb{N}$ respectively.

Consider the following oracle Turing machine M.

---

input $\boldsymbol{x}$
oracle $\boldsymbol{f}$
for $i := 1$ to l **do**
  $u_i := M_i^{\boldsymbol{f}}(\boldsymbol{x})$
end for
output $M_H^{\boldsymbol{f}}(u_1, \ldots, u_l)$

It is clear that M computes F.

In the for loop of M, the ith iteration uses space at most $P_i(|\boldsymbol{f}|, |\boldsymbol{x}|)$. The total space used in the for loop is

$$O\left(\sum_{i=1}^{l} P_i(|\boldsymbol{f}|, |\boldsymbol{x}|)\right).$$

The lengths of $u_1 = G_1(\boldsymbol{f}, \boldsymbol{x})$, $\ldots$, and $u_l = G_l(\boldsymbol{f}, \boldsymbol{x})$ are bounded by $P_1(|\boldsymbol{f}|, |\boldsymbol{x}|)$, $\ldots$, and $P_l(|\boldsymbol{f}|, |\boldsymbol{x}|)$ respectively. So the space use in the simulation of M_H is bounded by

$$P_H(|\boldsymbol{f}|, P_1(|\boldsymbol{f}|, |\boldsymbol{x}|), \ldots, P_l(|\boldsymbol{f}|, |\boldsymbol{x}|)).$$

The total space used in the computation of F is

$$O\left(P_H(|\boldsymbol{f}|, P_1(|\boldsymbol{f}|, |\boldsymbol{x}|), \ldots, P_l(|\boldsymbol{f}|, |\boldsymbol{x}|)) + \sum_{i=1}^{l} P_i(|\boldsymbol{f}|, |\boldsymbol{x}|)\right),$$

which is a second-order polynomial in $|\boldsymbol{f}|$ and $|\boldsymbol{x}|$ as both $|\boldsymbol{f}|$ and $|\boldsymbol{x}|$ are arbitrary.

Expansion Let $G \in \mathrm{p}_2\mathrm{space}$. Let M_G be the oracle Turing machine for G and let P_G be the corresponding second-order polynomial that bounds the space for M_G.

Let F be defined from G by expansion, i.e., for all $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y}$,

$$F(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y}) = G(\boldsymbol{f}, \boldsymbol{x}).$$

Consider the following oracle Turing machine M.

---

input $\boldsymbol{x}$, $\boldsymbol{y}$
oracle $\boldsymbol{f}$, $\boldsymbol{g}$
output $M_G^{\boldsymbol{f}}(\boldsymbol{x})$

It is clear that for all $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y}$

$$M^{\boldsymbol{f},\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{y}) = G(\boldsymbol{f}, \boldsymbol{x}) = F(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y}).$$

It is easy to verify that the second-order polynomial P defined by

$$P(|\boldsymbol{f}|, |\boldsymbol{g}|, |\boldsymbol{x}|, |\boldsymbol{y}|) = 2P_G(|\boldsymbol{f}|, |\boldsymbol{x}| + |\boldsymbol{y}|)$$

bounds the space of M.

Limited recursion on notation

Let $G, H, K \in \mathrm{p_2space}$. Let M_G, M_H, M_K be the oracle Turing machines that computes G, H, K respectively. Let P_G, P_H, P_K be their corresponding space bound respectively.

Let F be defined from G, H, K by limited recursion on notation.

Consider the following oracle Turing machine M.

---

input $\boldsymbol{x}$, w
oracle $\boldsymbol{f}$
$u := M_G^{\boldsymbol{f}}(\boldsymbol{x})$
for $i := 1$ to $|w| - 1$ **do**
  $u := M_H^{\boldsymbol{f}}(\boldsymbol{x}, w[0..i], u)$
end for
output u

Clearly, Turing machine M computes F (using the iterative expansion of the recursion). Now, we show that M runs in space that is bounded by a second-order polynomial.

The third line of code in M uses space bounded by $P_G(|\boldsymbol{f}|, |\boldsymbol{x}|)$.

At ith iteration of the for loop, the fifth line in M computes the value of $F(\boldsymbol{f}, \boldsymbol{x}, w[0..i])$ and uses space bounded by $P_H(|\boldsymbol{f}|, |F(\boldsymbol{f}, \boldsymbol{x}, w[0..i-1])|, |w[0..i]|)$. By the restriction in the definition of limited recursion on notation and the monotonicity of P_K, at any time during the computation

$$|F(\boldsymbol{f}, \boldsymbol{x}, w[0..i-1])| \leq P_K(|\boldsymbol{f}|, |\boldsymbol{x}|, |w|).$$

Thus the space used at ith iteration of the for loop is bounded by

$$P_H(|\boldsymbol{f}|, P_K(|\boldsymbol{f}|, |\boldsymbol{x}|, |w|), |w[0..i]|).$$

The auxiliary space used for the for loop is bounded by $c \cdot \log |w| \leq c \cdot |w|$, where $c > 0$ is some universal constant. The total amount of space used by M is bounded by

$$|w| \cdot P_H(|\boldsymbol{f}|, P_K(|\boldsymbol{f}|, |\boldsymbol{x}|, |w|), |w[0..i]|) + c \cdot |w|,$$

which is a second-order polynomial in $|\boldsymbol{f}|$, $|\boldsymbol{x}|$, and $|w|$. □

5 Adequacy for Measure

The general theory of resource-bounded measurability and measure developed in [15] consists of the basic definitions, reviewed below, and proofs that the resulting Δ-measure and measure in $R(\Delta)$ have the fundamental properties of a measure (e.g., additivity, measurability of measure-0 sets, etc.). The main shortcoming of the list-of-examples approach is evident in these proofs: Each time that a functional is asserted to be Δ-computable, it is *incumbent on the reader to check* that this holds for each of the infinitely many resource bounds Δ in the list.

Our main task in the present section is to re-prove these theorems in a more satisfactory manner. Our proofs here assume only that Δ is a resource bound, as defined in section 3, and they *explicitly* prove that the relevant functionals are Δ-computable, using only the axioms (closure properties) defining resource bounds.

To put the matter simply, the proofs in [15] are measure-theoretically rigorous, but their generality is tedious (for the conscientious reader) and limited. Our contribution here is to make these proofs and the scope of their validity explicit. For this reason, the proofs given in the present section focus on the Δ-computability of various type-two functionals, referring to [15] for the non-problematic, measure-theoretic parts of the proofs. We first review the definitions necessary for the development of a resource-bounded measure.

A *probability measure* on $\mathbf{C}$ is a function $\nu : \{0,1\}^* \to [0,1]$ such that $\nu(\lambda) = 1$ and, for all $w \in \{0,1\}^*$, $\nu(w) = \nu(w0) + \nu(w1)$. For strings $v, w \in \{0,1\}^*$, if $\nu(w) > 0$, we write $\nu(v|w)$ for the conditional probability of v given w. The *uniform probability measure* is μ such that $\mu(w) = 2^{-|w|}$ for all $w \in \{0,1\}^*$.

Let ν be a probability measure on $\mathbf{C}$. A ν*-martingale* is a function $d : \{0,1\}^* \to [0,\infty)$ with the property that for all $w \in \{0,1\}^*$,

$$d(w)\nu(w) = d(w0)\nu(w0) + d(w1)\nu(w1).$$

We use $\mathbf{1}$ for the *unit martingale* defined by $\mathbf{1}(w) = 1$ for all $w \in \{0,1\}^*$, which is a ν-martingale for every probability measure ν.

Definition 15. *Let ν be a probability measure. Let d be a ν-martingale. Let $A \subseteq \{0,1\}^*$. We say that d* covers *A if there is an $n \in \mathbb{N}$ such that $d(A[0..n-1]) \geq 1$. We say that d* succeeds *on A if $\limsup_{n\to\infty} d(A[0..n-1]) = \infty$ We say that d* succeeds strongly *on A if $\liminf_{n\to\infty} d(A[0..n-1]) = \infty$ The* set covered *by d (the* unitary success set*) is $S^1[d] = \{A \mid d \text{ covers } A\}$. The* success set *of d is $S^\infty[d] = \{A \mid d \text{ succeeds on } A\}$. The* strong success set *of d is $S^\infty_{\mathrm{str}}[d] = \{A \mid d \text{ succeeds strongly on } A\}$.*

We use real-valued functions (probability measures, martingales, etc.) on discrete domains of natural numbers $\mathbb{N}$ and strings $\{0,1\}^*$ extensively. Let D be a discrete domain. A *computation* of a function $f : D \to \mathbb{R}$ is a function $\hat{f} : \mathbb{N} \times D \to \mathbb{Q}$ such that, for all $r \in \mathbb{N}$ and $x \in D$, $|\hat{f}(r,x) - f(x)| \leq 2^{-r}$. In this expression, r may be thought of as the *precision* parameter of the computation. For such a function f, there is a unique computation $\hat{f}$ of f such that $\hat{f}_r(x) = a \cdot 2^{-r}$ for some integer a for all $r \in \mathbb{N}$ and $x \in D$. We call this particular $\hat{f}$ the *canonical computation* of f. Whenever a function f is involved as a parameter in the of a type-2 functional, the type-2 computation of the functional operates on the canonical computation, $\hat{f}$.

Definition 16 (Lutz [15]). *Let Δ be a resource bound. A Δ-*probability measure *on $\mathbf{C}$ is a probability measure ν on $\mathbf{C}$ such that ν is Δ-computable and there is a Δ-computable function $l : \mathbb{N} \to \mathbb{N}$ such that, for all $w \in \{0,1\}^*$, $\nu(w) = 0$ or $\nu(w) \geq 2^{-l(|w|)}$. We say that ν is* weakly positive*, if ν has the latter property.*

Definition 17 (Lutz [14,15]). *A* constructor *is a function $\delta : \{0,1\}^* \to \{0,1\}^*$ such that $x \mathrel{\underset{\neq}{\sqsubseteq}} \delta(x)$ for all $x \in \{0,1\}^*$. The* result *of δ is the unique language $R(\delta)$ such that $\delta^k(\lambda) \sqsubseteq R(\delta)$. If Δ is a resource bound, then the* result class *$R(\Delta)$ of Δ is the set of all languages $R(\delta)$ such that $\delta \in \Delta$.*

The martingale splitting operators defined by Lutz [15] are instrumental in developing the general theory of resource-bounded measurability and measure in complexity classes.

Definition 18 (Lutz [15]). *Let* X^+ *and* X^- *be disjoint subsets of* $\mathbf{C}$*, then a* ν-splitting operator *for* (X^+, X^-) *is a functional* $\Phi : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$*, such that* $\Phi(r, d) = (\Phi_r^+(d), \Phi_r^-(d))$ *has the following properties for all* $r \in \mathbb{N}$ *and* $d \in \mathcal{D}_\nu$*.*

(i) $X^+ \cap S^1[d] \subseteq S^1[\Phi_r^+(d)]$,
(ii) $X^- \cap S^1[d] \subseteq S^1[\Phi_r^-(d)]$,
(iii) $\Phi_r^+(d)(\lambda) + \Phi_r^-(d)(\lambda) \le d(\lambda) + 2^{-r}$.

If Δ *is a resource bound, a* Δ*-*ν*-splitting operator for* (X^+, X^-) *is a* ν*-splitting operator for* (X^+, X^-) *that is* Δ*-computable. Let* $X \subseteq \mathbf{C}$*. A* Δ*-*ν*-measurement of* X *is a* Δ*-*ν*-splitting operator for* (X^+, X^-)*. A* ν*-measurement of* X *in* $R(\Delta)$ *is a* Δ*-*ν*-splitting operator for* $(R(\Delta) \cap X^+, R(\Delta) - X)$*. If* Φ *is a* ν*-splitting operator, then we write* $\Phi_\infty^+ = \inf_{r\in\mathbb{N}} \Phi_r^+(\mathbf{1})(\lambda)$, $\Phi_\infty^- = \inf_{r\in\mathbb{N}} \Phi_r^-(\mathbf{1})(\lambda)$.

We now can define the resource-bounded measurabilities.

Definition 19. *A set* $X \subseteq \mathbf{C}$ *is* ν*-measurable in* $R(\Delta)$*, and we write* $X \in \mathcal{F}^\nu_{R(\Delta)}$*, if there exists a* ν*-measurement* Φ *of* X *in* $R(\Delta)$*. In this case, the* ν*-measure of* X *in* $R(\Delta)$ *is the real number* $\nu(X|R(\Delta)) = \Phi_\infty^+$*. (*$\nu(X|R(\Delta))$ *does not depend on the choice of* Φ *[15].)*

Definition 20. *A set* $X \subseteq \mathbf{C}$ *is* Δ*-*ν*-measurable, and we write* $X \in \mathcal{F}^\nu_\Delta$*, if there exists a* Δ*-*ν*-measurement* ϕ *of* X*. In this case, the* Δ*-*ν*-measure of* X *is the real number* $\nu_\Delta(X) = \Phi_\infty^+$*. (*$\nu_\Delta(X)$ *does not depend on the choice of* Φ *[15].)*

In the rest of this paper, we refer to [15] liberally whenever a claim was already proved.

Theorem 7 (Measure Conservation Theorem). *Let* Δ *be a resource bound. If* $w \in \{0,1\}^*$ *and* d *is a* Δ*-*ν*-martingale such that* $\mathbf{C}_w \cap R(\Delta) \subseteq S^1[d]$*, then* $d(\lambda) \ge \nu(w)$*.*

Proof. We define a functional

$$E : \mathcal{D}_\nu \times \mathbb{N} \times \{0,1\}^* \to (\{0,1\}^* \to \{0,1\}^*)$$

that maps ν-martingales to constructors.

Let $a : \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ be such that $a(x, m) = |x| + m + 2$. It is clear that a is BFF.

Let E be such that for all $d \in \mathcal{D}_\nu$, $m \in \mathbb{N}$, $w \in \{0,1\}^*$, and $x \in \{0,1\}^*$,

$$E(d, m, w)(x) = \begin{cases} w & \text{if } x \sqsubsetneq w \\ x0 & \text{if } \hat{d}_{a(x,m)}(x0) \le \hat{d}_{a(x,m)}(x1) \text{ and not } x \sqsubsetneq w \\ x1 & \text{otherwise,} \end{cases}$$

Note that $\hat{d}$ is the canonical computation of d and it is clear that E is BFF and $E(d, m, w)$ is a constructor.

Let $w \in \{0,1\}^*$ and let d be a Δ-ν-martingale such that $d(\lambda) < \nu(w)$. Then for every prefix $w' \sqsubseteq w$, there exists a constant $m \in \mathbb{N}$ such that $d(w) \le 1 - 2^{1-m}$ [15]. Let $\delta = E(d, m, w)$. By the proof of Lemma 3.4 in [15], $R(\delta) \notin S^1[d]$. Since $d \in \Delta$ and Δ is a resource bound hence closed under BFF, $\delta \in \Delta$. □

Lemma 2. *Let* ADD : $\mathcal{D}_\nu \times \mathcal{D}_\nu \to \mathcal{D}_\nu$ *be such that for all* $d', d'' \in \mathcal{D}_\nu$, $\mathrm{ADD}(d', d'') = d' + d''$*. Then* ADD $\in$ BFF.

Proof. We write d for $\mathrm{ADD}(d', d'')$. Then

$$\hat{d}_r(w) = \hat{d'}_{r+1}(w) + \hat{d''}_{r+1}(w).$$

Note that addition of two rational numbers in binary expansion is BFF. Since $|\hat{d'}_{r+1}(w) - d'(w)| \leq 2^{-r-1}$ and $|\hat{d''}_{r+1}(w) - d''(w)| \leq 2^{-r-1}$, $|\hat{d}_r(w) - (d'(w) + d''(w))| \leq 2^{-r}$ and $\hat{d}$ is the canonical computation of d. Therefore ADD is BFF. □

Lemma 3 (Lutz [15]). *Let Δ be a resource bound. Let $X \subseteq \mathbf{C}$. If Φ and Ψ are ν-measurements of X in $R(\Delta)$, then for all $j, k \in \mathbb{N}$,*

$$\Phi_j^+(\mathbf{1})(\lambda) + \Psi_k^+(\mathbf{1})(\lambda) \geq 1.$$

Proof. Assume the hypothesis, let $j, k \in \mathbb{N}$, and let

$$d = \mathrm{ADD}(\Phi_j^+(\mathbf{1}), \Psi_k^+(\mathbf{1})).$$

Since 1 is BFF, both Φ and Ψ are Δ, and Δ is a resource bound and closed under BFF, $d \in \Delta$. The rest of the proof is identical to the proof of Lemma 4.1 in [15]. □

Lemma 4 (Lutz [15]). *Let $X \subseteq \mathbf{C}$ and let Δ be a resource bound.*

1. *If X is Δ-ν-measurable, then X is ν-measurable in $R(\Delta)$ and $\nu(X|R(\Delta)) = \nu_\Delta(X)$.*
2. *X is ν-measurable in $R(\Delta)$ if and only if $X \cap R(\Delta)$ is ν-measurable in $R(\Delta)$, in which case $\nu(X \mid R(\Delta)) = \nu(X \cap R(\Delta) \mid R(\Delta)$.*

Theorem 8 (Lutz [15]). *Let Δ be a resource bound. Let $X \subseteq \mathbf{C}$.*

1. *If X is ν-measurable in $R(\Delta)$, then $\nu(X|R(\Delta))$ is Δ-computable.*
2. *If X is Δ-ν-measurable, then $\nu_\Delta(X)$ is Δ-computable.*

Proof. We prove 1, since 2 follows by Lemma 4.

Let S_ν be the set of all ν-splitting operators. Note that $\mathbf{1}$ is BFF $\subseteq \Delta$. Let Φ be a ν-measurement for X in $R(\Delta)$. Then $\Phi \in \Delta$. Since Δ is a resource bound and closed under BFF, $r \mapsto \hat{\Phi}^+_{r,r}(\mathbf{1})$ is in Δ. Let

$$f(r) = \hat{\Phi}^+_{r+1,r+1}(\mathbf{1})(\lambda).$$

Then $f \in \Delta$ and by the proof of Theorem 4.7 in [15], f is the canonical computation of the real value of $\nu(X|R(\Delta))$. □

We now proceed towards the proof that cylinders are measurable. First, we prove lemmas that are useful for the proof of Theorem 9.

Lemma 5 (Regularity Lemma). *Let Δ be a resource bound. There is a functional*

$$\Lambda : \mathcal{D}_\nu \to \mathcal{D}_\nu$$

with the following properties.

1. *For all $d \in \mathcal{D}_\nu$, $\Lambda(d)$ is a regular ν-martingale such that $\Lambda(d)(\lambda) = d(\lambda)$ and $S^1[d] \subseteq S^1[\Lambda(d)]$.*
2. *$\Lambda(\mathbf{1}) = \mathbf{1}$.*

3. If ν is a Δ-probability measure on $\mathbf{C}$*, then Λ is Δ-computable.*

The proof of the regularity lemma proceeds by defining a type-2 functional. We establish the computability properties of this functional, first.

Lemma 6 (Pasting Lemma). *Let Δ be a resource bound, p, q be natural numbers and $f, g, k : \mathbb{R}^p \to \mathbb{R}^q$ be uniformly continuous, Δ computable functions. Let $h : \mathbb{R}^p \to \mathbb{R}^q$ be the piecewise function defined by*

$$h(\boldsymbol{r}) = \begin{cases} f(\boldsymbol{r}) & \textit{if } k(\boldsymbol{r}) \geq 0 \\ g(\boldsymbol{r}) & \textit{otherwise.} \end{cases}$$

If h is continuous everywhere, then it is Δ-computable.

Proof. Let $\hat{f}, \hat{g}, \hat{k}$ be the respective computations of appropriate types of f, g, and k. Let m be the maximum of the modulus functions of f, g and k. (For the modulus function to be type-1, we need uniform continuity.) We define the following functional $\hat{h}$ and prove that it is a Δ- computation of h.

$$\hat{h}(\hat{\boldsymbol{r}}, 0^n) = \begin{cases} \hat{f}(\hat{\boldsymbol{r}})(0^{m(n)+1}) & \text{if } \hat{k}(\hat{\boldsymbol{r}})(0^{m(n)+1}) \geq 0 \\ \hat{g}(\hat{\boldsymbol{r}})(0^{m(n)}+1) & \text{otherwise.} \end{cases}$$

If $k(\boldsymbol{r})$ and $\hat{k}(\hat{\boldsymbol{r}})(0^{m(n)+1})$ are of the same sign, then $|h(r) - \hat{h}(\hat{\boldsymbol{r}}, 0^n)| < 2^{-n}$ by the property of the witnesses $\hat{f}$ and $\hat{g}$.

If $k(\boldsymbol{r}) < 0$ and $\hat{k}(\hat{\boldsymbol{r}})(0^{m(n)+1}) \geq 0$, we have that $\hat{k}(\hat{\boldsymbol{r}})(0^{m(n)}) < 2^{-(n+1)}$ by the approximation property of k. We also have $\hat{h}(\hat{\boldsymbol{r}}, 0^n) = \hat{f}(\hat{\boldsymbol{r}})(0^{m(n)+1})$, and $h(\boldsymbol{r}) = g(\boldsymbol{r})$. Thus,

$$\begin{aligned} |\hat{h}(\hat{\boldsymbol{r}}, 0^n) - h(\boldsymbol{r})| &= |\hat{h}(\hat{\boldsymbol{r}}, 0^n) - g(\boldsymbol{r})| \\ &= |\hat{f}(\hat{\boldsymbol{r}})(0^{m(n)+1}) - g(\boldsymbol{r})|. \end{aligned}$$

Since k changes sign in the $2^{-[m(n)+1]}$ neighborhood of $\boldsymbol{r}$, there is a point r_1 in it where $k(\boldsymbol{r_1}) = 0$ (This follows from the fact that $\mathbb{R}^q$ is a connected set.). Since h is continuous at r_1, we can conclude $f(r_1) = g(r_1)$.

Thus,

$$\begin{aligned} |\hat{f}(\hat{\boldsymbol{r}})(0^{m(n)+1}) - g(\boldsymbol{r})| &\leq |\hat{f}(\hat{\boldsymbol{r}}(0^{m(n)+1}) - f(\boldsymbol{r_1})| + |f(\boldsymbol{r_1}) - g(\boldsymbol{r})| \\ &= |\hat{f}(\hat{\boldsymbol{r}}(0^{m(n)+1}) - f(\boldsymbol{r_1})| + |g(\boldsymbol{r_1}) - g(\boldsymbol{r})| \\ &\leq 2^{-(n+1)} + 2^{-(n+1)}. \end{aligned}$$

The case when $k(\boldsymbol{r}) \geq 0$ but $\hat{k}(\hat{\boldsymbol{r}})(0^{m(n)}) < 0$ is similar. □

We can now confirm that the "Robin Hood function" is Δ-time computable.

Lemma 7. *Let Δ be a resource bound and let α be a Δ-computable real number. Then the function $m_\alpha : \mathbb{R}^2 \to \mathbb{R}$ defined by*

$$m_\alpha = \alpha s + (1 - \alpha)t$$

is a Δ-computable uniformly continuous function.

Lemma 8. *Let α be a Δ-computable real number in $(0,1)$, H_α be the half-plane*

$$H_\alpha = \{(x,y) | \quad x, y \in \mathbb{R}, m_\alpha(x,y) \geq 1\}$$

and $D_\alpha = [0,\infty)^2 \cup H_\alpha$. Then the "Robin Hood function" $rh_\alpha : D_\alpha \to [0,\infty)^2$ defined by

$$rh_\alpha(s,t) = \begin{cases} (s,t) & \textit{if } (s,t) \in [0,1]^2 \\ (m_\alpha(s,t), m_\alpha(s,t)) & \textit{if } m_\alpha(s,t) \geq 1 \\ \left(1, \frac{m_\alpha(s-1,t)}{1-\alpha}\right) & \textit{if } m_\alpha(s,t) < 1, s \geq 1, t \geq 0 \\ \left(\frac{m_\alpha(s,t-1)}{\alpha}, 1\right) & \textit{if } m_\alpha(s,t) < 1, t \geq 1, s \geq 0. \end{cases}$$

is Δ-computable.

Proof. The Robin Hood function is a continuous piecewise linear mapping from the Euclidean plane to itself, and we have $\alpha \in \Delta$. Hence each component of the Robin Hood function is Δ-computable. The regions of the Robin Hood function are defined by the lines

1. $y = 1$.
2. $x = 1$.
3. $m_\alpha(s,t) - 1 = 0$.
4. $y = 0$ and $x = 0$.

All of these are linear functions, hence all of them are Δ-computable uniformly continuous functions. Inside D_α, $0 \leq s \leq 1$ if and only if $(1-s)s \geq 0$. Define $\textsc{inside} : D_\alpha \to \mathbb{R}$ by

$$\textsc{inside}(s,t) = \min\{(1-s)s, (1-t)t\}.$$

We conclude that $\textsc{inside}(s,t) \geq 0$ if and only if $(s,t) \in [0,1]^2$. Also, in D_α, a point (s,t) is inside the triangle defined by the lines $x = 1$, $y = 0$ and $m_\alpha(x,y) = 1$ if and only if $(s-1)t(1-m_\alpha(s,t)) \geq 0$. It follows that each of the following functions is Δ-computable, by Lemma 6.

$$rh_\alpha(s,t) = \begin{cases} (s,t) & \text{if INSIDE}(s,t) \geq 0 \\ rh1_\alpha(s,t) & \text{otherwise.} \end{cases}$$

$$rh1_\alpha(s,t) = \begin{cases} (m_\alpha(s,t), m_\alpha(s,t)) & \text{if } m_\alpha(s,t) \geq 1 \\ rh2_\alpha(s,t) & \text{otherwise.} \end{cases}$$

$$rh2_\alpha(s,t) = \begin{cases} \left(1, \frac{m_\alpha(s-1,t)}{1-\alpha}\right) & \text{if } (s-1)t(1-m_\alpha(s,t)) \geq 0 \\ rh3_\alpha & \text{otherwise.} \end{cases}$$

$$rh3_\alpha(s,t) = \begin{cases} \left(\frac{m_\alpha(s,t-1)}{\alpha}, 1\right) & \text{if } (t-1)s(1-m_\alpha(s,t)) \geq 0 \\ (1,1) & \text{otherwise.} \end{cases}$$

Thus the Robin-Hood function is computable. □

The following essential properties of the Robin Hood function rh_α are routine to verify.

1. The transformation rh_α is a continuous, piecewise linear mapping from D_α into $[0,\infty)^2$.
2. The transformation rh_α preserves α-weighted averages, i.e., $m_\alpha(rh_\alpha(s,t)) = m_\alpha(s,t)$ for all $(s,t) \in D_\alpha$.
3. The transformation rh_α maps H_α into $[1,\infty)^2$. That is, if the average $m_\alpha(s,t)$ is at least 1, then rh_α "steals from the richer and gives to the poorer" of s and t so that both $rh_\alpha^{(0)}(s,t)$ and $rh_\alpha^{(1)}(s,t)$ are at least 1.
4. For all $(s,t) \in D_\alpha$, $rh_\alpha^{(0)}(s,t) \geq \min\{1,s\}$ and $rh_\alpha^{(1)}(s,t) \geq \min\{1,t\}$. That is, the transformation rh_α never "steals" more than the excess above 1.
5. The transformation rh_α leaves points of $[0,1]^2$ unchanged.

A ν-martingale is *regular* if, for all $v,w \in \{0,1\}^*$, if $\nu(v) \geq 1$ and $v \sqsubseteq w$, then $\nu(w) \geq 1$. It is often technically convenient to have a uniform means of ensuring that martingales are regular. The following lemma provides such a mechanism. Let Δ be a resource bound, as specified in section 2, and let ν be a probability measure on $\mathbf{C}$.

Proof (Proof of Lemma 5). Using the Robin Hood function, we define the functional $\Lambda : \mathcal{D}_\nu \to \mathcal{D}_\nu$ as follows. For $d \in \mathcal{D}_\nu$, we define the ν-martingale $\Lambda(d)$ by the following recursion. (In all clauses, $w \in \{0,1\}^*$ and $b \in \{0,1\}$.)

(i) $\Lambda(d)(\lambda) = d(\lambda)$.
(ii) If $\nu(w) = 0$ or $\nu(wb \mid w) \in \{0,1\}$, then $\Lambda(d)(wb) = \Lambda(d)(w)$.
(iii) If $\nu(w) > 0$ and $0 < \nu(wb \mid w) < 1$, then

$$\Lambda(d)(wb) = rh^{(b)}_{\nu(w0|w)}(g_0(w), g_1(w)),$$

where $g_b(w) = \Lambda(d)(w) - d(w) + d(wb)$.

Let $\hat{d}$ be a computation of d and $\hat{\nu} : \{0,1\}^* \times 0^{\mathbb{N}} \to \mathbb{Q}$ be the function testifying that ν is Δ computable. Since ν is Δ-computable, we have a function $l : \mathbb{N} \to \mathbb{N}$ in Δ such that for every string w, $\nu(w) > 2^{-l(|w|)}$. The functional $\hat{\Lambda} : \mathcal{D}_\nu \times \{0,1\}^* \times 0^{\mathbb{N}} \to \mathbb{Q}$ is a BFF computation of Λ.

1. $\hat{\Lambda}(\hat{d})(\lambda, 0^n) = \hat{d}(\lambda, 0^n)$.
2. If $\hat{\nu}(w) < 2^{-l(|w|)}$ or $\hat{\nu}(w0, 0^n) < 2^{-l|w0|}$ or $\hat{\nu}(w1, 0^n) < 2^{-l|w1|}$, then $\hat{\Lambda}(d)(wb, 0^n) = d(wb, 0^n)$.
3. Otherwise, $\hat{\Lambda}(\hat{d})(wb, 0^n) = \hat{rh}^{(b)}_{\hat{\nu}(w0|w)}(\hat{g}_0(w, 0^n), g_1(w, 0^n))$.

Note that if ν is a strongly positive probability measure, then $\nu(b|w) = 1$ if and only if $\nu(w\overline{b}|w) = 0$. Assuming $\nu(w) > 0$, we have that $\nu(w\overline{b}|w) = 0$ if and only if $\nu(w\overline{b}) = 0$, i. e. $\nu(\overline{b}) < 2^{-l(|w|)}$. Thus step 2 correctly approximates step 2 of Λ.

It is now routine (if tedious) to verify that Λ has the desired properties. □

Lemma 9. *Let type-2 functional $B : (\{0,1\}^* \to \mathbb{R}) \times (\mathbb{N} \to \mathbb{N}) \to (\{0,1\}^* \times \{0,1\}^* \to \mathbb{R})$ be such that for every weakly positive probability measure $\nu : \{0,1\}^* \to [0,1]$, $l : \mathbb{N} \to \mathbb{N}$, and $w, v \in \{0,1\}^*$,*

$$B(\nu, l)(w, v) = \begin{cases} \nu(w|v) & \textit{if } v \sqsubseteq w \textit{ and } \nu(w) \geq 2^{-l(|w|)} \\ 1 & \textit{if } w \sqsubseteq v \textit{ and } \nu(w) \geq 2^{-l(|w|)} \\ 0 & \textit{otherwise.} \end{cases}$$

Then B is BFF *over all weakly positive probability measure ν and all $l : \mathbb{N} \to \mathbb{N}$.*

Theorem 9 (Lutz [15]). *Let Δ be a resource bound. If ν is a Δ-probability measure on $\mathbf{C}$, then for each $w \in \{0,1\}^*$, the cylinder $\mathbf{C}_w$ is Δ-ν-measurable, with $\nu_\Delta(\mathbf{C}_w) = \nu(w)$.*

Proof. Assume the hypothesis, and let $w \in \{0,1\}^*$. We prove this lemma in two cases.

Case 1: $\nu(w) = 0$. Let $\Phi : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$ be such that for each $r \in \mathbb{N}$, $d \in \mathcal{D}_\nu$, and $v \in \{0,1\}^*$,

$$\Phi^+_r(d)(v) = \begin{cases} 1 & \text{if } w \sqsubseteq v \\ 0 & \text{otherwise,} \end{cases}$$
$$\Phi^-_r(d)(v) = d(v).$$

It is clear that Φ is BFF and hence Δ-computable and that Φ is a ν-splitting operator. By the proof of Lemma 4.8 in [15], for all $r \in \mathbb{N}$ and $d \in \mathcal{D}_\nu$, Φ has the following properties: (i) $S^1[d] \cap \mathbf{C}_w \subseteq \mathbf{C}_w = S^1[\Phi^+_r(d)]$; (ii) $S^1[d] - \mathbf{C}_w \subseteq S^1[d] = S^1[\Phi^-_r(d)]$; (iii) $\Phi^+_r(d)(\lambda) + \Phi^-_r(d)(\lambda) = d(\lambda)$. Therefore, Φ is a Δ-ν-measurement of $\mathbf{C}_w$. It can be shown that $\nu_\Delta(w) = 0$ [15]. Note that $w \neq \lambda$, since $\nu(\lambda) = 1$ and $\nu(w) = 0$.

Case 2: $\nu(w) > 0$. Let $\Psi' : \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$ be such that for each $d \in \mathcal{D}_\nu$ and $v \in \{0,1\}^*$,

$$\Psi'^+(d)(v) = \begin{cases} d(w)\nu(w|v) & \text{if } v \sqsubseteq w \\ d(v) & \text{if } w \sqsubseteq v \\ 0 & \text{otherwise,} \end{cases}$$
$$\Psi'^-(d)(v) = d(v) - \Psi'^+(d)(v).$$

Since ν is a Δ-probability measure, there exists $l : \mathbb{N} \to \mathbb{N}$ such that l is Δ-computable and for all $w \in \{0,1\}^*$, $\nu(w) = 0$ or $\nu(w) \geq 2^{-l(|w|)}$. Then

$$\Psi'^+(d)(v) = \begin{cases} d(w)B(\nu, l)(w, v) & \text{if } v \sqsubseteq w \\ d(v) & \text{if } w \sqsubseteq v \\ 0 & \text{otherwise,} \end{cases}$$

where B is the functional defined in Lemma 9. Note that Ψ' is BFF over type-1 input d and $\nu(\cdot|\cdot)$. By Lemma 9, we have that Ψ' is BFF over probability measure ν and ν-martingale d.

Let $\Psi : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$ be such that $\Psi(r, d) = \Psi'(\Lambda(d))$, where Λ is the functional from the Regularity Lemma. Since ν is a Δ-probability measure, Λ is Δ-computable. Since Ψ' is BFF and $d, \nu, l, \Lambda \in \Delta$, Ψ is Δ-computable since Δ is closed under BFF. The rest of the proof is to establish that Ψ is a Δ-ν-measurement of $\mathbf{C}_w$ and $\nu_\Delta(\mathbf{C}_w) = \nu(w)$, which follows directly from the proof of Lemma 4.8 in [15].

Definition 21. *Let $R \subseteq \mathbf{C}$. An* algebra *on R is a collection $\mathcal{F}$ of subsets of $\mathbf{C}$ with the following properties.*

(i) $R \in \mathcal{F}$.
(ii) *If $X \in \mathcal{F}$, then $X^c \in \mathcal{F}$.*
(iii) *If $X, Y \in \mathcal{F}$, then $X \cup Y \in \mathcal{F}$.*

If $\mathcal{F}$ is an algebra on R, then a subalgebra *of $\mathcal{F}$ on R is a set $\mathcal{E}$ that is also an algebra on R.*

Theorem 10 (Lutz [15]). *Let Δ be a resource bound. $\mathcal{F}^\nu_{R(\Delta)}$ is an algebra on $R(\Delta)$. For $X, Y \in \mathcal{F}^\nu_{R(\Delta)}$, we have*

$$\nu(X^c|R(\Delta)) = 1 - \nu(X|R(\Delta))$$

and

$$\nu(X \cup Y|R(\Delta)) = \nu(X|R(\Delta)) + \nu(Y|R(\Delta)) - \nu(X \cap Y|R(\Delta)).$$

Note that this lemma does not have any computability requirement on ν. It is possible that $\mathcal{F}^\nu_{R(\Delta)} = \{\varnothing, \mathbf{C}\}$.

Proof. Let Φ be a ν-measurement of X in $R(\Delta)$. Let

$$\Psi : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$$

be such that

$$\Psi(r, d) = (\Phi^-_r(d), \Phi^+_r(d)).$$

Since Φ is Δ-computable and Ψ can be defined from Φ with projection and composition, Ψ is Δ-computable. Then by Theorem 4.12 in [15], Ψ is a ν-measurement of X^c in $R(\Delta)$, i.e., $X^c \in \mathcal{F}^\nu_{R(\Delta)}$ and $\nu(X^c|R(\Delta)) = 1 - \nu(X|R(\Delta))$.

Now, let $X, Y \in \mathcal{F}^\nu_{R(\Delta)}$ and let Φ and Ψ be ν-measurements of X and Y, respectively, in $R(\Delta)$. For each $a, b \in \{+, -\}$, let

$$\Theta[ab] : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu$$

by

$$\Theta[ab](r, d) = \Psi^b_{r+2}(\Phi^a_{r+1}(d)).$$

Note that $\Theta[ab]$ is defined from Ψ and Φ by BFF. Therefore, for each $a, b \in \{+, -\}$, $\Theta[ab]$ is Δ-computable, since Δ is closed under BFF. Let

$$\Theta[X \cap Y] = (\Theta[++] + \Theta[-+], \Theta[+-] + \Theta[--]),$$
$$\Theta[X \cup Y] = (\Theta[++] + \Theta[-+], \Theta[+-] + \Theta[--]).$$

Then $\Theta[X \cap Y]$ and $\Theta[X \cup Y]$ are BFF in $\Theta[ab]$ ($a, b \in \{+, -\}$), $\Theta[X \cap Y], \Theta[X \cup Y]$ are Δ-computable. Then by the proof of Theorem 4.12 in [15], $\Theta[X \cap Y]$ and $\Theta[X \cup Y]$ are ν-measurements of $X \cap Y$ and $X \cup Y$ in $R(\Delta)$ respectively, and thus $X \cap Y$ and $X \cup Y$ are ν-measurable in $R(\Delta)$. The second identity in the theorem also follows from Theorem 4.12 in [15]. $\square$

Corollary 1 (Lutz [15]). *Let Δ be a resource bound. Let $X, Y \in \mathcal{F}^{\nu}_{R(\Delta)}$.*

1. *If $X \cap Y \cap R(\Delta) = \varnothing$, then*
$$\nu(X \cup Y \mid R(\Delta)) = \nu(X \mid R(\Delta)) + \nu(Y \mid R(\Delta)).$$
2. *If $X \cap R(\Delta) \subseteq Y$, then $\nu(X \mid R(\Delta)) \leq \nu(Y \mid R(\Delta))$.*

Theorem 11. *Let Δ be a resource bound. $\mathcal{F}^{\nu}_{\Delta}$ is an algebra over $\mathbf{C}$. For $X, Y \in \mathcal{F}^{\nu}_{\Delta}$, we have*
$$\nu_{\Delta}(X^c) = 1 - \nu_{\Delta}(X)$$
and
$$\nu_{\Delta}(X \cup Y) = \nu_{\Delta}(X) + \nu_{\Delta}(Y) - \nu_{\Delta}(X \cup Y).$$

Proof. The proof is similar to that of Theorem 10. □

Definition 22. *Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^{\nu}_{R(\Delta)}$ on $R(\Delta)$. Then $\mathcal{F}$ is ν-*complete *on $R(\Delta)$ if, for all $X, Y \subseteq \mathbf{C}$, if $X \subseteq Y \in \mathcal{F}$ and $\nu(Y|R(\Delta)) = 0$, then $X \in \mathcal{F}$.*

Definition 23. *Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^{\nu}_{\Delta}$ on $\mathbf{C}$. Then $\mathcal{F}$ is Δ-ν-*complete *if, for all $X, Y \subseteq \mathbf{C}$, if $X \subseteq Y \in \mathcal{F}$ and $\nu_{\Delta}(Y) = 0$, then $X \in \mathcal{F}$.*

Theorem 12. *The algebra $\mathcal{F}^{\nu}_{R(\Delta)}$ is ν-complete on $R(\Delta)$ and the algebra $\mathcal{F}^{\nu}_{\Delta}$ is Δ-ν-complete.*

Proof. We prove the case with $\mathcal{F}^{\nu}_{R(\Delta)}$. The other case is similar. Let $X \subseteq Y \in \mathcal{F}^{\nu}_{R(\Delta)}$ and $\nu(Y|R(\Delta)) = 0$. Let Φ be a ν-measurement of Y in $R(\Delta)$. Let
$$\Psi : \mathbb{N} \times \mathcal{D}_{\nu} \to \mathcal{D}_{\nu} \times \mathcal{D}_{\nu}$$
be such that
$$\Psi(r, d) = (\Phi^{+}_{r}(\mathbf{1}), d).$$
Note that Ψ is BFF in Φ and $\mathbf{1}$. Since Φ is Δ-computable and Δ is a resource bound, Ψ is also Δ-computable. By the proof of Theorem 4.16 in [15], Ψ is a ν-measurement of X in $R(\Delta)$ and thus $X \in \mathcal{F}^{\nu}_{R(\Delta)}$. □

We cannot hope to have countable additivity in this theory [15], however we have the following additivity property over uniformly computable sequences.

Definition 24 (Lutz [15]). *Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^{\nu}_{R(\Delta)}$ on $R(\Delta)$. A Δ-sequence in $\mathcal{F}$ is a sequence $(X_k \mid k \in \mathbb{N})$ of sets $X_k \in \mathcal{F}$ for which there exists a Δ-computable functional*
$$\Phi : \mathbb{N} \times \mathbb{N} \times \mathcal{D}_{\nu} \to \mathcal{D}_{\nu} \times \mathcal{D}_{\nu}$$
such that, for each $k \in \mathbb{N}$, Φ_k is a ν-measurement of X_k in $R(\Delta)$.

Definition 25 (Lutz [15]). *Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^{\nu}_{\Delta}$ on $\mathbf{C}$. A Δ-sequence in $\mathcal{F}$ is a sequence $(X_k \mid k \in \mathbb{N})$ of sets $X_k \in \mathcal{F}$ for which there exists a Δ-computable functional*
$$\Phi : \mathbb{N} \times \mathbb{N} \times \mathcal{D}_{\nu} \to \mathcal{D}_{\nu} \times \mathcal{D}_{\nu}$$
such that, for each $k \in \mathbb{N}$, Φ_k is a Δ-ν-measurement of X_k.

Definition 26 (Lutz [15]). *A functional*

$$\Phi : \mathbb{N} \times \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$$

is Δ-modulated if the sequences $(\Phi^a_{k,r}(d)(w) \mid k \in \mathbb{N})$, for $a \in \{+,-\}$, $r \in \mathbb{N}$, $d \in \mathcal{D}_\nu$, and $w \in \{0,1\}^*$, are uniformly Δ-convergent. Equivalently, Φ is Δ-modulated if there is a Δ-computable functional

$$\Gamma : \mathbb{N} \times \mathbb{N} \times \mathcal{D}_\nu \times \{0,1\}^* \to \mathbb{N}$$

such that, for all $a \in \{+,-\}$, $t, r \in \mathbb{N}$, $d \in \mathcal{D}_\nu$, $w \in \{0,1\}^*$, and $k \geq \Gamma(t,r,d,w)$,

$$\left|\Phi^a_{k,r}(d)(w) - \Phi^a_{\infty,r}(d)(w)\right| \leq 2^{-t}$$

where the limit $\Phi^a_{\infty,r}(d)(w) = \lim_{k\to\infty} \Phi^a_{k,r}(d)(w)$ is implicitly assumed to exist.

Definition 27 (Lutz [15]). *Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^\nu_{R(\Delta)}$ on $R(\Delta)$.*

1. *A union Δ-sequence in $\mathcal{F}$ is a sequence $(X_k \mid k \in \mathbb{N})$ of sets $X_k \in \mathcal{F}$ for which there exists a Δ-modulated functional Φ such that $\Phi^+_{k,r}(d)(w)$ is nondecreasing in k, $\Phi^-_{k,r}(d)(w)$ is nonincreasing in k, and Φ testifies that $\left(\bigcup_{j=0}^{k-1} X_j \mid k \in \mathbb{N}\right)$ is a Δ-sequence in $\mathcal{F}$.*
2. *An intersection Δ-sequence in $\mathcal{F}$ is a sequence $(X_k \mid k \in \mathbb{N})$ of sets $X_k \in \mathcal{F}$ for which there exists a Δ-modulated functional Φ such that $\Phi^+_{k,r}(d)(w)$ is nonincreasing in k, $\Phi^-_{k,r}(d)(w)$ is nondecreasing in k, and Φ testifies that $\left(\bigcap_{j=0}^{k-1} X_j \mid k \in \mathbb{N}\right)$ is a Δ-sequence in $\mathcal{F}$.*
3. *$\mathcal{F}$ is closed under Δ-unions if $\bigcup_{k=0}^\infty X_k \in \mathcal{F}$ for every union Δ-sequence $(X_k \mid k \in \mathbb{N})$ in $\mathcal{F}$.*
4. *$\mathcal{F}$ is closed under Δ-intersections if $\bigcap_{k=0}^\infty X_k \in \mathcal{F}$ for every intersection Δ-sequence $(X_k \mid k \in \mathbb{N})$ in $\mathcal{F}$.*

Lemma 10. *Let Δ be a resource bound. Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^\nu_{R(\Delta)}$ on $R(\Delta)$. If $(X_k \mid k \in \mathbb{N})$ is a Δ-sequence in $\mathcal{F}$ and $\nu(X_k \mid R(\Delta)) = 0$ for all $k \in \mathbb{N}$, then $(X_k \mid k \in \mathbb{N})$ is a union Δ-sequence in $\mathcal{F}$.*

Proof. Assume the hypothesis, and let Φ be a witness that $(X_k \mid k \in \mathbb{N})$ is a Δ-sequence in $\mathcal{F}$. Let

$$\Psi : \mathbb{N} \times \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$$

be such that

$$\Psi_{k,r}(d) = \left(\sum_{j=0}^{k} \Phi^+_{j,j+r+1}(\mathbf{1}), d\right)$$

for all $k, r \in \mathbb{N}$ and $d \in \mathcal{D}_\nu$. Note that the bounded sum $\sum_{j=0}^k$ is BFF. Then Ψ is Δ-computable since Δ is a resource bound. Together with the proof of Lemma 4.18 in [15], we have that Ψ testifies that $(X_k \mid k \in \mathbb{N})$ is a union Δ-sequence. □

Lemma 11. *Let Δ be a resource bound. Let $\mathcal{F}$ be a subalgebra of $\mathcal{F}^\nu_{R(\Delta)}$ on $R(\Delta)$. Then a sequence $(X_k \mid k \in \mathbb{N})$ is a union Δ-sequence in $\mathcal{F}$ if and only if the complemented sequence $(X^{\mathrm{c}}_k \mid k \in \mathbb{N})$ is an intersection Δ-sequence in $\mathcal{F}$. Thus $\mathcal{F}$ is closed under Δ-unions if and only if $\mathcal{F}$ is closed under Δ-intersections.*

Proof. Note that for any ν-splitting operator Φ, $\Phi = (\Phi^+, \Phi^-)$ is a Δ-ν-splitting operator if and only if (Φ^-, Φ^+) is a Δ-ν-splitting operator, since Δ is closed under BFF. The rest of the proof is identical to that of Lemma 4.19 in [15]. □

Theorem 13 (Lutz [15]). *Let Δ be a resource bound.*

1. *$\mathcal{F}^{\nu}_{R(\Delta)}$ is closed under Δ-unions and Δ-intersections.*
2. *If $(X_k \mid k \in \mathbb{N})$ is a union Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$, then*

$$\nu(\cup_{k=0}^{\infty} X_k \mid R(\Delta)) \leq \sum_{k=0}^{\infty} \nu(X_k \mid R(\Delta)),$$

with equality if the sets $X_0, X_1, \ldots$ are disjoint.

3. *If $(X_k \mid k \in \mathbb{N})$ is a union Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$ with each $X_k \subseteq X_{k+1}$, then*

$$\nu(\cup_{k=0}^{\infty} X_k \mid R(\Delta)) = \lim_{k\to\infty} \nu(X_k \mid R(\Delta)).$$

4. *If $(X_k \mid k \in \mathbb{N})$ is an intersection Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$ with each $X_{k+1} \subseteq X_k$, then*

$$\nu\left(\cap_{k=0}^{\infty} X_k \mid R(\Delta)\right) = \lim_{k\to\infty} \nu\left(X_k \mid R(\Delta)\right).$$

Proof. Let $(X_k \mid k \in \mathbb{N})$ be a union Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$, with the functional Φ as witness, and let $X = \bigcup_{k=0}^{\infty} X_k$. Fix a functional Γ testifying that Φ is Δ-modulated, and define a functional

$$\Theta : \mathbb{N} \times \mathcal{D}_\nu \to \mathcal{D}_\nu \times \mathcal{D}_\nu$$

such that for all $r \in \mathbb{N}$ and $d \in \mathcal{D}_\nu$,

$$\Theta(r, d) = (\Phi^{+}_{\infty,r+1}, \Phi^{-}_{m,r+1}(d)),$$

where $m = \Gamma(r+1, r+1, d, \lambda)$. Note that Θ is BFF in Φ and Γ. By our choice of Φ and Γ, Θ is in BFF(Δ) and thus Δ-computable. Then Θ is a ν-measurement of X in $R(\Delta)$ by the proof of Theorem 4.20 in [15]. Therefore $X \in \mathcal{F}^{\nu}_{R(\Delta)}$. This, together with Lemma 11, proves part 1 of the theorem.

Part 2 of this theorem then follows from the proof of the part 2 of Theorem 4.20 in [15].

To prove part 3 of the theorem, let $(X_k \mid k \in \mathbb{N})$ be a union Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$ with functional Φ as a witness such that $X_k \subseteq X_{k+1}$ for each $k \in \mathbb{N}$. Therefore, Φ also testifies that $\left(\bigcup_{j=0}^{k-1} X_j \mid k \in \mathbb{N}\right)$ is a Δ-sequence in $\mathcal{F}$.

Let $Y_0 = X_0$ and, for each $k \in \mathbb{N}$, let $Y_{k+1} = X_{k+1} - X_k$. Note that since $X_k \subseteq X_{k+1}$,

$$\bigcup_{j=0}^{k} X_j = X_k = \bigcup_{j=0}^{k} Y_j.$$

Therefore Φ also testifies that $\left(\bigcup_{j=0}^{k-1} Y_j \mid k \in \mathbb{N}\right)$ is a Δ-sequence in $\mathcal{F}$ and hence testifies that $(Y_k \mid k \in \mathbb{N})$ is a union Δ-sequence in $\mathcal{F}^{\nu}_{R(\Delta)}$. The rest of the proof of part 3 is identical to that of Theorem 4.20 in [15], which can be argued using part 2 of this theorem, Theorem 10, and taking a limit over k of the measures $\nu(X_k \mid R(\Delta))$.

The proof of part 4 is identical to that in the proof of Theorem 4.20 in [15]. □

References

1. E. Allender and M. Strauss. Measure on small complexity classes with applications for BPP. In *Proceedings of the 35th Symposium on Foundations of Computer Science*, pages 807–818, 1994.

2. K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics, pages 1–47. Marcel Dekker, New York, N.Y., 1997.
3. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, second edition, 1995.
4. H. Buhrman and L. Torenvliet. On the structure of complete sets. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, pages 118–133, 1994.
5. H. Buhrman and L. Torenvliet. Complete sets and structure in subrecursive classes. In *Proceedings of Logic Colloquium '96*, volume 12 of *Lecture Notes in Logic*, pages 45–78, 1998.
6. A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science II*, pages 24–30. North–Holland, 1965.
7. J. J. Dai. An outer-measure approach for resource-bounded measure. *Theory of Computing Systems*, 45(1):64–73, 2009.
8. N. Danner and J. Royer. Adventures in time and space. *Logical Methods in Computer Science*, 3(1:9):1–53, 2007.
9. R. C. Harkins and J. M. Hitchcock. Dimension, halfspaces, and the density of hard sets. In *Proceedings of the 13th Annual International Computing and Combinatorics Conference*, volume 4598 of *Lecture Notes in Computer Science*, pages 129–139. Springer-Verlag, 2007.
10. J. M. Hitchcock. Resource-Bounded Measure Bibliography, http://www.cs.uwyo.edu/ ∼jhitchco/bib/rbm.html (current January, 2009).
11. R. Impagliazzo and B. Kapron. Logics for reasoning about cryptographic constructions. *J. Comput. Syst. Sci.*, 72(2):286–320, 2006.
12. R. Irwin, B. Kapron, and J. Royer. On characterizations of basic feasible functionals, part i. *Journal of Functional Programming*, 11(1):117–153, 2001.
13. B. Kapron and S. Cook. A new characterization of type 2 feasibility. *SIAM Journal on Computing*, 25:117–132, 1996.
14. J. H. Lutz. Almost everywhere high nonuniform complexity. *J. Comput. Syst. Sci.*, 44(2):220–258, 1992.
15. J. H. Lutz. Resource-bounded measure. In *Proceedings of the 13th IEEE Conference on Computational Complexity*, pages 236–248, 1998.
16. J. H. Lutz and E. Mayordomo. Measure, stochasticity, and the density of hard languages. *SIAM Journal on Computing*, 23(4):762–779, 1994.
17. J. H. Lutz and E. Mayordomo. Twelve problems in resource-bounded measure. *Bulletin of the European Association for Theoretical Computer Science*, 68:64–80, 1999.
18. E. Mayordomo. Measuring in PSPACE. In *Proceedings of the 7th International Meeting of Young Computer Scientists*, volume 6 of *Topics in Computer Science*, pages 93–100. Gordon and Breach, 1994.
19. K. Mehlhorn. Polynomial and abstract subrecursive classes. *Journal of Computer and System Science*, 12:147–178, 1976.
20. P. Moser. Martingale families and dimension in P. *Theoretical Computer Science*, 400(1–3):46–61, 2008.
21. P. Moser. Resource-bounded measure on probabilistic classes. *Information Processing Letters*, 106(6):241–245, 2008.
22. A. Pavan. Comparison of reductions and completeness notions. *SIGACT News*, 34(2):27–41, June 2003.
23. Plato. *Dialogues*, volume 1. Yale University Press, 1989.
24. D. B. Thompson. Subrecursiveness: machine independent notions of computability in restricted time and storage. *Mathematical Systems Theory*, 6:3–15, 1972.
25. K. Weihrauch. *Teilklassen primitiv-rekursiver Wortfunktionen.* PhD thesis, Rheinische Friedrich-Wilhelms-Universitt Bonn, 1973.