Towards a Logical Foundation of Randomized Computation (Thesis Abstract)*

Melissa Antonelli[1]

[1]HIIT Helsinki Institute for Information Technology, Pietari Kalmi katu, 5, 00560, Helsinki, Finland.

Abstract

Interactions between logic and computer science have been deeply investigated in the last century, but, surprisingly, the study of probabilistic computation was only marginally touched by such fruitful interchanges. The goal of my doctoral project was precisely that of start bridging this gap by developing logical systems corresponding to specific aspects of randomized computation and, thus, by generalizing standard achievements to the probabilistic realm. To do so, the key ingredient is the introduction of new, measure-sensitive quantifiers associated with quantitative interpretations. Concretely, the dissertation is tripartite. The first part concerns counting complexity and its main result is the proof that classical counting propositional logic provides a purely logical characterization of Wagner's hierarchy [1]. In the second part, which focusses on programming language theory, we present a probabilistic Curry-Howard correspondence [2] between the intuitionistic version of our counting propositional logic and the typed probabilistic λ-calculus with counting quantifiers. Finally, we consider the relationship between arithmetic and computation by introducing a quantitative extension of the language of Peano arithmetic able to formalize basic results from probability theory. This language is also our starting point to define *randomized* bounded arithmetic and, so, to generalize canonical results by Buss [3].

Keywords

Randomized Computation, Logical Foundations of Computer Science, Probability Logic, Reasoning about Uncertainty

1. Introduction

Historically, determinacy was certainly one of the defining features of standard computational models: given an algorithm and an input, the sequence of computation steps is uniquely determined. In the second half of the XX century, this assumption started to be relaxed in different ways and randomized algorithms were introduced for the first time – where a *randomized algorithm* is a process which can evolve probabilistically so that, given an input, the computation it performs may lead to different outcomes, each associated with a certain probability. Such a more flexible design makes this computational model a very efficient and powerful tool, with several applications in computer science (CS, for short) and technology.

Starting from this, my Ph.D. dissertation has been motivated by two main considerations. On

BEWARE 2023: Joint Workshop, AIxIA 2023, November 6-9, 2023, Rome, Italy,

*The author thanks her Ph.D. thesis supervisor U. Dal Lago and co-supervisor P. Pistone for their constant guidance and crucial help: all the results presented here are part of the joint work with them. The author is grateful to Helsinki Institute for Information Technology - HIIT for supporting her work since 2023. Finally, the author wish to thank the anonymous reviewers for helpful comments.

melissa.antonelli@helsinki.fi (M. Antonelli)

the one hand, since its early appearance in the 1950s, probabilistic computation has become ubiquitous in several fast-growing areas of CS and, by now, related, abstract models – as probabilistic Turing machines (PTM, for short), stochastic automata or randomized λ-calculi – have been deeply investigated in the literature. On the other, there exist deep and mutual interactions linking logic and theoretical computer science (TCS, for short), and, in the past, the development and study of computation theory and models has considerably benefitted from them. Yet, randomized computation was only marginally touched by such fruitful interchanges and, so far, it has not found a precise logical counterpart. Such a missing connection looks even more striking nowadays, due to the increasing pervasiveness of probability in many relevant fields of information technology, from AI and statistical learning to cryptography, approximate computing and robotics. The global purpose of my doctoral thesis consisted in laying the foundation for a uniform approach to bridge the quoted gap. To do so, the key ingredient is the introduction of a family of new logics, the language of which includes non-standard quantifiers "measuring" the probability of their argument formula, and associated with inherently quantitative semantics.

2. The Importance of Being Randomized

As said, randomized algorithms are powerful tools with numerous applications in different fields and technologies. Generally speaking, these are crucial when dealing with *uncertain* information and *partial* knowledge, namely for all systems acting in realistic contexts – think, for example, of driverless cars [4] or of computer vision modelling [5]. Notably, in some areas probabilistic models have become even more than optional; for instance in cryptography, where secure encryption schemas are probabilistic [6].

On Logical Foundations of Computer Science. The existence of several and deep interactions between logic and TCS is not accidental, but rooted in the intimate correspondence connecting these disciplines. In fact, even the formal appearance of the *science of computing* was essentially motivated by foundational studies in mathematics and logic, that had defined the context in which this subject took its first steps. Later on, the back and forth between logic and CS has strongly influenced the development of both and, today, numerous areas of IT – as programming language theory [2], verification [7] and database theory [8], computational and descriptive complexity [9, 10], just to quote a few – has effectively benefitted from their mutual dialogue. As Siekmann wrote, "[i]n many respects, logic provides computer science with both a unifying foundational framework and a tool for modeling" [11, p. 17]. Indeed, several aspects of computer *science* are intrinsically related with logic, as shown by a variety of seminal results, e.g. [12, 13, 14, 9]. The other side of the coin is the existence of numerous *concrete* exchanges between these disciplines: while the growing importance of IT has guided and stimulated many advances in logic, logical tools have extensive applications in CS and technology – from software and hardware verification to the modelling of interactive or multi-agent systems, from the study of relational databases to argumentation theory.

Probabilistic Computation. Probabilistic computational models have been widely investigated in the last few years, and are nowadays pervasive in almost every areas of CS. As seen, the idea of relaxing the notion of algorithm from *purely deterministic* to *probabilistic* appeared early in the history of modern computability theory. Intuitively, as anticipated, a randomized algorithm is nothing but an algorithm involving random processes – typically corresponding to "flipping a coin" – as part of its procedure. While in deterministic computation, for every input, the algorithm $\mathcal{A}$ produces (at most) one output, in randomized computation, given an input, the algorithm $\mathcal{A}_{\mathcal{R}}$ returns a set of outputs, each associated with a probability:

$$[\![\mathcal{A}]\!] : \mathbb{N} \rightharpoonup \mathbb{N} \qquad \rightsquigarrow \qquad [\![\mathcal{A}_{\mathcal{R}}]\!] : \mathbb{N} \to \mathcal{D}_{\mathbb{N}}.$$

In this way, algorithms have enabled efficient solutions to numerous problems [15], becoming essential in different areas. As a consequence, several probabilistic (formal) models were introduced: from PTMs [16, 17] and stochastic automata [18, 19] to probabilsitic λ-calculi [20]. At this point, randomized algorithms and programs are widespread, steering disciplines like robotics, AI, verification and security coding, computer vision and NLP:

> The last decade has witnessed a tremendous growth in the area of randomized algorithms. During this period, randomized algorithms went from being a tool in computational number theory to finding widespread applications in many types of algorithms. Two benefits of randomization have spearheaded this growth: simplicity and speed. [15, p. ix]

***Reasoning* About Uncertainty.** In particular, the use of randomized models have spread in discipline involving uncertain domain – that is, in all disciplines *realistically* interacting with "the world". For instance, in agent systems (whether artificial or not) reasoning is processed and decisions are made on the ground of partial information obtained from the environment and the background knowledge.[1] Clearly, in such contexts, simplifications are needed and "probabilistic thinking" appears as a formidable tool for decision making and learning processing [21]. These new, concrete demands also led to the first attempts to analyze probabilistic reasoning *in a formal way* and to the introduction of a few *logical* systems, starting in 1986 with Nilsson's pioneering proposal:

> Because many artificial intelligence applications require the ability to reason with uncertain knowledge, it is important to seek appropriate generalizations of logic from this case. [22, p. 71]

In the following years, new probability logics, inspired by [22], were presented and developed in the context of modal logic [23, 24, 25]. Nowadays, different formal systems to deal with probabilistic and uncertain reasoning have been defined, even basing on alternative (non-modal) approaches, for example via non-monotonic and fuzzy logics or due to direct numerical representations.[2]

[1] Probabilistic models become fundamental in AI research from the 1970s-1980s on. For further details on the "main phases" in the history of this discipline, see e.g. [5].

[2] A detailed overview of logics for probability can be found e.g. in [21] or in [26, pp. 91ff.].

3. Towards Logical Foundations of Randomized Computation

As anticipated, interchanges between logic and computation are numerous and well-studied. Yet, in the randomized setting, such a deep correspondence has only been sparsely investigated. One crucial peculiarity, when switching to probabilistic algorithms, is that, in this case, behavioral properties, like termination or equivalence, have an *inherently quantitative* nature, that is a computation terminates *with a given probability* and a program might simulate a desired function *up to some probability of error* – think, for instance, to probabilistic primality tests or learning algorithms. Then, the central question is:

can such quantitative properties be studied within a logical system?

In my Ph.D. dissertation a positive answer is given, at least for the specific aspects of the interaction between quantitative logics and randomized computation it focusses on. As we shall briefly see, the turning point of our approach consists in considering new *quantitative* logics able to express probability in a natural way.

3.1. Relating Logic and Randomized Computation

Concretely, we generalized a few standard results linking logic and computation to the probabilistic realm.

Complexity Theory. As it is well-known, classical propositional logic and computational complexity are strongly connected. Indeed, checking the satisfiability of **PL**-formulas is the paradigmatic $\mathbf{NP}$-complete problem [9], while the language of classical tautologies is $\mathbf{coNP}$-complete. In the early 1970s, Meyer and Stockmeyer also showed that, when switching to *quantified* propositional logic (**QPL**, for short), the full polynomial hierarchy ($\mathbf{PH}$, for short) can be captured by a *single* logical system and that each level in it is characterized by the validity of **QPL**-formulas (in PNF), with the corresponding number of quantifier alternations. Nonetheless, when moving to the probabilistic framework, such a plain correspondence seems lost since no analogous *logical* counterpart is known to relate in a similar way to the counting classes and hierarchy, as introduced by Valiant [27] and Wagner [1]:

$$\text{polynomial hierarchy} : \mathbf{QPL} \quad \Longleftrightarrow \quad \text{counting hierarchy} : ?$$

In the first part of the dissertation, a counting propositional system, called **CPL**, is introduced. This logic is basically a generalization of **PL** capable of expressing that a formula is true *with a given probability* [28, 29]. Then, **CPL** is shown to be strongly related to counting computation and classes, being the probabilistic counterpart of **QPL** [30, 28, 31]. Indeed, its counting quantifiers can be naturally seen as "quantitative" versions of standard propositional ones. Our main result here is the *purely logical* characterization of Wagner's hierarchy via complete problems defined in terms of **CPL**-formulas.

Programming Language Theory. Traditionally, the Curry-Howard correspondence (CHC, for short) relates intuitionistic **PL** and the simply-typed λ-calculus, but in the last fifty years

this correspondence was shown to hold in other and more sophisticated contexts too. Meanwhile, randomized λ-calculi [20] and associated type systems were introduced, sometimes also guaranteeing desirable forms of termination [32]. Yet, they are not so-to-say "logically oriented" and no (probabilistic) CHC [2] is known for them:

$$\text{simply typed } \lambda_{\to} : \text{intuitionistic } \mathbf{PL} \iff \text{randomized } \lambda\text{-calculi} : ?$$

In the second part of the thesis, two new ingredients are introduced to define the first probabilistic version of the quoted correspondence. On the one hand, we consider the intuitionistic version of univariate **CPL**, called $\mathbf{iCPL}_0$, and show it able to capture quantitative behavioral properties. On the other, we define a "counting"-typed randomized λ-calculus. Its untyped part is strongly inspired by the probabilistic event λ-calculus introduced in [33], while its types are defined mimicking counting quantifiers. Finally, we establish a (static and dynamic) correspondence in the style of Curry and Howard between these two systems [34, 31].

Arithmetic and Computation Theory. The theory of (deterministic) computation and arithmetic are linked by deep results coming from logic and recursion theory – for example, Gödel's arithmetization [35] or realizability [36] or the *Dialectica* interpretation [37]. Indeed, the language of arithmetic is able to express many interesting properties of algorithms and, due to the relation between totality (of functions) and termination (of algorithms), several issues in computation theory can be analyzed in the framework of arithmetic. Also in this context, when switching to the probabilistic realm, no theory can be found to relate to randomized computation as Peano Arithmetic (**PA**, for short) does in the deterministic case:

$$\text{deterministic computation} : \mathbf{PA} \iff \text{probabilisitic computation} : ?$$

In the third part of the dissertation, we present a quantitative extension of the language of **PA**, called **MQPA**, which allows us to formalize basic results from probability theory that are not expressible in **PA**, for example the so-called infinite monkey theorem or the random walk theorem. This language is also proved to be actually connected to randomized computation as we establish the first probabilistic version of Gödel's arithmetization [33], namely it is shown that any *random* function can be expressed by a formula of **MQPA**.

In addition, this language is at the basis of our study of randomized bounded theories [38]. One of the central motivations for the development of bounded arithmetics – i.e., subsystems of **PA** the language of which includes functions with specific growth rate together with bounded quantifiers, and in which induction is (variously) limited – was their connection with computational complexity. As it is clear that not all computable functions are *feasibly* computable, bounded theories become essential to characterize interesting (feasible) complexity classes in terms of families of arithmetic formulas. Specifically, in 1986 Buss proved that poly-time computable functions precisely correspond to those functions which are Σ^b_1-definable in a given bounded theory, called $\mathbf{S}^1_2$ [3]. Although this fact is very insightful, no similar result was established in the probabilistic framework:

$$\text{deterministic classes} : \mathbf{BA} \iff \text{probabilistic classes} : ?$$

Inspired by the language **MQPA**, in the third part of the thesis we also introduce a *randomized* bounded theory, called $\mathbf{RS}^1_2$, enabling us to logically capture relevant probabilistic classes, as $\mathbf{BPP}$ [39, 40].

3.2. From Evaluating to Measuring

Counting quantifiers are quantifiers of the form $\mathbf{C}^q_X$ or $\mathbf{D}^q_X$ and capable of expressing probabilities within a logical language. These quantifiers not only determine the *existence* of a satisfying assignment, but rather count *how many* those assignments are. In this sense, counting quantifiers can be seen as a *quantitative* generalization of standard propositional ones:

$$(\forall X)F, (\exists X)F \qquad \rightsquigarrow \qquad \mathbf{C}^q_X F, \mathbf{D}^q_X F.$$

Intuitively, the **QPL**-formula $(\exists X)F$ says that there is an interpretation for X making F true. On the other hand, the (pseudo-)counting formula $\mathbf{C}^{1/2}_X F$ expresses that F has probability greater than $\frac{1}{2}$ of being true, i.e. that F is true for at least one half of all possible interpretations of X. Dually, $\mathbf{D}^{1/2}_X F$ says that the argument formula F has probability strictly smaller than $\frac{1}{2}$ of being true.

Remarkably such a generalization is made possible by switching from a truth-functional to a quantitative semantics, in which formulas are no more interpreted as single truth-values but as measurable sets of models:

$$[\![F]\!]_{\mathbf{QPL}} \in \{0, 1\} \qquad \rightsquigarrow \qquad [\![F]\!]_{\mathbf{CPL}} \subseteq 2^{\mathbb{N}}.$$

So, while (the truth of) an existentially-quantified formula of **QPL** – for instance, $(\exists X)(\exists Y)(X \wedge Y)$ – gives us information about the *existence* of a model for $X \wedge Y$, counting-quantified formulas tell us something about the *number* of these satisfying valuations. For example, the (pseudo-)counting formula $\mathbf{C}^{1/4}_{X,Y}(X \wedge Y)$ says not only that there is a model for $X \wedge Y$, but also that *at least* one out of four possible interpretations of the argument formula is a satisfying one. It is in this way that counting logical systems allow us to formally represent and study quantitative aspects of probabilistic computation in an innovative way.

Notably, our counting *propositional* logics are natural tools to represent stochastic events in a straightforward way,[3] but, as predictable, their expressive power is still quite limited. This has led us to the generalization of the notion of counting quantifier and to the definition of the extended language **MQPA**, which is nothing but the language of first-order arithmetic endowed with second-order measure quantifiers and associated with a Borel semantics.

References

[1] K. Wagner, The Complexity of Combinatorial Problems with Succinct Input Representation, Acta Informatica 23 (1986) 325–356.

[2] M. Sorensen, P. Urzyczyn, Lectures on the Curry-Howard Isomorphism, volume 149, Elsevier, 2006.

[3] S. Buss, Bounded Arithmetic, Ph.D. thesis, Princeton University, 1986.

[4] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2006.

[3] For further details, see [29].

[5] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[6] S. Goldwasser, S. Micali, Probabilistic Encryption, Journal of Computer and System Sciences 28 (1984) 279–299.
[7] M. Thornton, R. Drechsler, D. Miller, Logic Verification, Springer, 2001, pp. 201–230.
[8] E. Codd, Relational Completeness of Data Base Sublanguages, in: Proc. 6th Courant Computer Science Symposium, 1972, pp. 65–98.
[9] S. Cook, The Complexity of Theorem-Proving Procedures, in: Proc. Symposium on Theory of Computing (STOC), 1971, pp. 151–158.
[10] N. Immerman, Descriptive Complexity, Springer, 1999.
[11] J. Siekman, Computational Logic, in: J. Siekmann (Ed.), Handbook of the History of Logic: Computational Logic, volume 9, Elsavier, 2014, pp. 15–30.
[12] A. Turing, On Computable Numbers, with an Application to the *Entscheidungsproblem*, in: Proc. London Mathematical Society, volume 42, 1936, pp. 230–265.
[13] A. Church, S. Kleene, Formal Definitions in the Theory of Ordinal Numbers, Fundamenta Mathematicae 28 (1936) 11–21.
[14] W. Howard, The Formulae-as-Types Notion of Construction, in: J. Seldin, J. Hindley (Eds.), To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, Academic Press, 1980, pp. 479–490.
[15] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[16] E. Santos, Probabilistic Turing Machines and Computability, Proc. American Mathematical Society 22 (1969) 704–710.
[17] J. Gill, Computational Complexity of Probabilistic Turing Machines, in: Proc. Symposium on Theory of Computing (STOC), 1974, pp. 91–95.
[18] M. O. Rabin, Probabilistic Automata, Information and Computation 6 (1963) 230–245.
[19] R. Segala, A Compositional Trace-Based Semantics for Probabilistic Automata, in: Proc. International Conference on Concurrency Theory (CONCUR), 1995, pp. 234–248.
[20] N. Saheb-Djaromi, Probabilistic LCF, in: A. Press (Ed.), Proc. International Symposium on Mathematical Foundations of Computer Science, 1978, pp. 154–165.
[21] J. Pearl, Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference, Elsavier, 1988.
[22] N. Nilsson, Probabilistic Logic, Artificial Intelligence 28 (1986) 71–87.
[23] F. Bacchus, Representing and Reasoning with Probabilistic Knowledge, MIT Press, 1990.
[24] R. Fagin, J. Halpern, N. Megiddo, A Logic for Reasoning about Probabilities, Information and Computation 87 (1990) 78–128.
[25] J. Halpern, Reasoning About Uncertainty, MIT Press, 2003.
[26] M. Antonelli, Towards a Logical Foundation of Randomized Computation, Ph.D. thesis, University of Bologna, Department of Computer Science and Engineering, 2023.
[27] L. Valiant, The Complexity of Computing the Permanent, Theoretical Computer Science 8 (1979) 189–201.
[28] M. Antonelli, U. Dal Lago, P. Pistone, On Counting Propositional Logic and Wagner's Hierarchy, Theoretical Computer Science (2023).
[29] M. Antonelli, Two Remarks on Counting Propositional Logic, in: Proc. Workshop on Bias, Risk, Explainability and the role of Logic and Logic Programming (BEWARE), AIxIA Conference, 2022, pp. 20–32.
[30] M. Antonelli, U. Dal Lago, P. Pistone, On Counting Propositional Logic and Wagner's Hierarchy, in: Proc. Italian Conference of Theoretical Computer Science (ICTCS), 2021, pp. 107–121.
[31] M. Antonelli, U. Dal Lago, P. Pistone, Towards Logical Foundations for Probabilistic Computation, Annals of Pure and Applied Logic (2023).
[32] C. Faggian, S. Ronchi della Rocca, Lambda Calculus and Probabilistic Computation, in: Proc. Symposium on Logic in Computer Science (LICS), 2019, pp. 1–13.
[33] U. Dal Lago, G. Guerrieri, W. Heijltjes, Decomposing Probabilistic Lambda-Calculi, in: Proc. Foundations of Software Science and Computation Structures (FoSSaCS), 2020, pp. 136–156.
[34] M. Antonelli, U. Dal Lago, P. Pistone, Curry and Howard Meet Borel, Proc. Symposium on Logic in Computer Science (LICS) (2022) 1–13.
[35] K. Gödel, Über Formal Unentscheidbare Sätze der *Principia Mathematica* und Verwandter Systeme, Monatsch. Math. Phys. 38 (1931) 173–178.
[36] G. Kreisel, Interpretation of Analysis by Means of Constructive Funcionals of Finite Types, in: A. Heyting (Ed.), Constructivity in Mathematics, North-Holland, 1959, pp. 101–128.
[37] K. Gödel, Über eine Bisher noch nicht Benützte Erweiterung des Finiten Standpunktes, Dialectica 12 (1958)

280–287.
[38] M. Antonelli, U. Dal Lago, D. Davoli, I. Oitavem, P. Pistone, Enumerating Erorr Bounded Polytime Algorithms Through Arithmetical Theories, in: Proc. Computer Science Logic (CSL), 2024.
[39] M. Antonelli, U. Dal Lago, D. Davoli, I. Oitavem, P. Pistone, Towards a Randomized Bounded Arithmetic, in: AILA - Book of Abstract, 2022.
[40] M. Antonelli, U. Dal Lago, D. Davoli, I. Oitavem, P. Pistone, Enumerating Erorr Bounded Polytime Algorithms Through Arithmetical Theories, in: Logic Colloquium 2023 - Book of Abstract, 2023, pp. 45–46.
[41] M. Antonelli, U. Dal Lago, P. Pistone, On Measure Quantifiers in First-Order Arithmetic, in: L. De Mol, M. F. Weiermann, A., D. Fernández-Duque (Eds.), Proc. Computability in Europe Conference (CiE), 2021, pp. 12–24.
[42] P. Billingsley, Probability and Measure, Wiley, 1995.

A. Outline of the Thesis

The Ph.D. dissertation *Towards a Logical Foundation of Randomized Computation* [26] was defended in July 2023 at the University of Bologna, Department of Computer Science and Engineering (DISI). Its main contributions concern three aspects of the interaction between *quantitative* logic and *probabilistic* computation. Accordingly, the thesis is divided into three parts. Each one is intended to be as self-contained as possible and the opening chapter always offers a bird's-eye view of the topic captioned in the corresponding part. It includes a brief historical overview and global motivations, together with an informal presentation of the results to be later considered.

- In the first part of the dissertation [26, Ch. 2-5], counting propositional logics are introduced and proved able to define complete (logical) problems for each level of Wagner's hierarchy. Specifically, the languages of univariate $\mathbf{CPL}_0$ and of multivariate **CPL** are presented in Chapters 3 and 4, respectively, together with the associated, quantitative semantics. These systems provide a natural formalism to represent stochastic events (see [26, Sec. 3.4]) and support a suitable proof-theoretical treatment (see [26, Sec. 3.3, Sec. 4.2]), in the form of sound and complete sequent calculi. The main result of this part is the *logical* characterization of $\mathbf{CH}$, as presented in Chapter 5.
- The second part of the thesis [26, Ch. 6-10] is devoted to our proposal of a probabilistic CHC. In Chapter 7, the intuitionistic version of counting propositional logic, called **iCPL**, is defined, while, in Chapter 8, the computational part of the correspondence, namely a slightly modified version of the probabilistic event λ-calculus by Dal Lago, Guerrieri and Heijltjes [33], is considered. The principal contribution here is the definition of a *probabilistic* CHC between a fragment of intuitionistic counting propositional logic and a counting-typed system able to express the probability of termination. This is presented in Chapter 9. Then, in Chapter 10, termination properties are further investigated by introducing a related intersection type system.
- Finally, in the third part of the dissertation [26, Ch. 11-13], a more general language, called **MQPA**, is introduced together with a quantitative semantics. This language extends that of **PA** via second-order measure quantifiers, which are not far from counting ones. In Chapter 12, it is shown that the expressive power of **MQPA** is effectively remarkable. Indeed, results from probability theory, which cannot be expressed in **PA**, can instead be properly formalized in it. Furthermore, it is proved that any recursive random function can be represented by a formula of **MQPA**. In Chapter 13, a new randomized bounded arithmetic theory is defined and it is established that the class of formulas which are Σ^b_1-representable in it is precisely that of poly-time random functions. Due to this result, an *arithmetical* characterization of $\mathbf{BPP}$ is provided.

All the quoted contributions are part of the joint work with Ugo Dal Lago and Paolo Pistone. Our research about randomized bounded theory, as presented in Section 14, was developed together with Davide Davoli and Isabel Oitavem. Crucial results from [26, Part I] have been partially presented in [30, 29, 28, 31]. The main contributions of [26, Part II] have been published in [34, 31]. The language **MQPA** and its connections with random functions, as introduced in [26, Part III], have been presented in [41], while results concerning randomized bounded theories [26, Ch. 13] have appeared (or will appear) in [39, 40, 38].

B. Counting and Measure-Quantified Logics in a Nutshell

In Section B.1, we introduce a few standard notions from basic probability theory which are needed to define the semantics of our counting and measure-quantified logics. In Section B.2, we present the language and semantics of univariate $\mathbf{CPL}_0$ and of multivariate **CPL**, while, in Section B.3, we deal with the language and semantics of the more expressive language **MQPA**.

B.1. Basic Probability Theory

In probability theory, an *outcome* or *point* ω is the result of a single execution of an experiment, the *sample space* Ω is the set of all possible outcomes, and an *event* E is a subset of Ω. Two events, say E_1 and E_2, are *disjoint* or *mutually exclusive* when they cannot happen at the same time, that is $E_1 \cap E_2 = \emptyset$. A class $\mathfrak{F}$ of subsets of Ω is said to be a *σ-field* or *σ-algebra* when (i.) it contains Ω, i.e. $\Omega \in \mathfrak{F}$, (ii) it is closed under complementation, i.e. if $E \in \mathfrak{F}$, then $\overline{E} \in \mathfrak{F}$, being $\overline{E}$ the complementation of E, and (iii) it is closed under countable union (and intersection). The largest σ-field on Ω is the power class 2^{Ω}, while the smallest one is $\{\emptyset, \Omega\}$. The *σ-field generated by* $\mathfrak{F}$, $\sigma(\mathfrak{F})$, is the smallest σ-algebra containing $\mathfrak{F}$. A *measurable space* is a pair $(\Omega, \mathfrak{F})$, where $\mathfrak{F}$ is a σ-algebra over Ω.

In the 1930s, Kolmogorov introduced the notion of *probability space*, together with the axioms for probability. A *probability measure* on a σ-field $\mathfrak{F}$, $\textsc{Prob}(\cdot)$, associates each event $E \in \mathfrak{F}$ with a number, $\textsc{Prob}(E)$, so that:

i. for each $E \in \mathfrak{F}$, $0 \leq \textsc{Prob}(E) \leq 1$,
ii. $\textsc{Prob}(\emptyset) = 0$ and $\textsc{Prob}(\Omega) = 1$,
iii. if $E_1, E_2, ... \in \mathfrak{F}$ is a sequence of disjoint events, then $\textsc{Prob}\big(\bigcup_{k=1}^{\infty} E_k\big) = \sum_{k=1}^{\infty} \textsc{Prob}(E_k)$.

Two events are *(stochastically) independent* when the occurrence of one does not affect the probability for the other to occur. In particular, given two disjoint events, say E_1 and E_2, $\textsc{Prob}(E_1 \cup E_2) = \textsc{Prob}(E_1) + \textsc{Prob}(E_2)$, while for two independent events, say E_1' and E_2', $\textsc{Prob}(E_1' \cap E_2') = \textsc{Prob}(E_1') \cdot \textsc{Prob}(E_2')$. A *probability space* $(\Omega, \mathfrak{F}, \textsc{Prob})$ is a mathematical object that provides a formal model for random processes and is made of:

- a *sample space*, Ω, which is the set of all possible outcomes,
- a *σ-field*, $\mathfrak{F}$, which is the set of events,
- a *probability measure*, Prob, which assigns to each event in $\mathfrak{F}$ a probability, i.e. a number between 0 and 1 satisfying the so-called Kolmogorov axioms.

In the following, we will mostly focus on a specific probability space such that $\Omega = 2^{\mathbb{N}}$, namely the set of all infinite sequences of random bits (i.e. coin tosses). Each such sequence is denoted as

$$\omega = \omega(1)\omega(2)\ldots,$$

with $\omega \in \Omega$, $i \in \mathbb{N}$, and $\omega(i) \in \{0, 1\}$. Each sequence ω can be interpreted as the result of infinitely flipping a coin.

Definition 1 (Cylinder of Rank n)**.** A *cylinder of rank n* is a set of the form

$$\mathscr{cyl}_H = \{\omega \mid \omega(1), ..., \omega(n) \in H\},$$

with $H \subset \{0, 1\}^n$.

Observe that when $H = \{(u_1, ..., u_n)\}$ is a singleton $\big($for $u_1, ..., u_n \in \{0, 1\}\big)$, an event $E = \{\omega \mid \omega(1), ..., \omega(n) = (u_1, ..., u_n)\}$, such that the first n repetitions of the experiment have outcomes $u_1, ..., u_n$ in sequence, is called a *thin cylinder*. We will be particularly interested in thin cylinders in which the only set defining H is made of one element $u_i = 1$.

Notation 1. For $i \in \mathbb{N}$, we denote *special thin cylinders* as follows:

$$Cyl(i) = \{\omega \mid \omega(i) = 1\}.$$

The class of cylinders of all ranks, which is a field [42, pp. 27-30], is denoted by $\mathscr{C}$, while $\sigma(\mathscr{C})$ indicates the σ-algebra generated by $\mathscr{C}$. It is thus possible to define a measure on it. Specifically, we use $\mu_{\mathscr{C}}$ to denote the unique probability measure over $\big(2^{\mathbb{N}}, \sigma(\mathscr{C})\big)$, such that for any $i \in \mathbb{N}$, $\mu_{\mathscr{C}}\big(Cyl(i)\big) = \frac{1}{2}$.

Definition 2 (Canonical Cylinder Measure $\mu_{\mathscr{C}}$)**.** Given $u \in \{0, 1\}$, let p_u denote the (non-negative and summing to 1) probability of getting u. Then, for any cylinder $\mathscr{cyl}_H$,

$$\mu_{\mathscr{C}}(\mathscr{cyl}_H) = \sum_{H} p_{u_1} \cdots p_{u_n},$$

the sum extending over all sequences $(u_1, \ldots, u_n) \in H$.

B.2. On Counting Propositional Logic

The core idea to define counting propositional logics is to extend standard languages by *measure-sensitive quantifiers* and to consider *quantitative* semantics in which formulas are no more interpreted as single truth values, but as measurable sets of satisfying valuations. When dealing with the most intuitive, univariate fragment $\mathbf{CPL}_0$, any formula, say F, is interpreted as the set $[\![F]\!] \subseteq 2^{\mathbb{N}}$ made of all maps $f \in 2^{\mathbb{N}}$ "making F true" $\big($and belonging to the standard Borel algebra over $2^{\mathbb{N}}$, $\mathscr{B}(2^{\mathbb{N}})\big)$. In particular, atomic propositions are interpreted as special cylinder sets of the form:

$$Cyl(i) = \{f \in 2^{\mathbb{N}} \mid f(i) = 1\},$$

for $i \in \mathbb{N}$, while non-atomic expressions are interpreted as standard operations of complementation, finite intersection and union. Since these sets are all measurable, and $\mathscr{B}(2^{\mathbb{N}})$ is endowed with a canonical probability measure, it makes sense to ask whether "F is true with probability *at least* q" or "F is true with probability *strictly smaller than* q". This is formalized by the notion of counting quantifier – i.e. by $\mathbf{C}^q$ or $\mathbf{D}^q$, for $q \in \mathbb{Q} \cap [0,1]$ –, inspired by Wagner's counting operator [1]. Intuitively, the formula $\mathbf{C}^q F$ (resp., $\mathbf{D}^q F$) expresses that F is satisfied by a portion of assignments greater (resp., strictly smaller) than q. For example, $\mathbf{C}^{1/2} F$ expresses the fact that F is satisfied by *at least half* of its valuations. In the more expressive counting propositional logic, **CPL**, relations between valuations of different groups of variables can be taken into account. Contextually, the corresponding quantitative semantics is subtler than that of $\mathbf{CPL}_0$ and the interpretation for counting-quantified formulas relies on some technical notions. Remarkably, in [30, 28, 31], sound and complete proof system(s) for both $\mathbf{CPL}_0$ and **CPL** are introduced.

Remark 1. There is a strong connection between (closed) formulas of $\mathbf{CPL}_0$ and (closed) formulas of **CPL** in which only one name occurs. Indeed, a translation which preserves validity can be easily defined to pass from ones to the others [26, Sec. 4.1].

B.2.1. Syntax and Semantics of $\mathbf{CPL}_0$

Definition 3 (Formulas of $\mathbf{CPL}_0$)**.** *Formulas of* $\mathbf{CPL}_0$ are defined by the grammar below:

$$F ::= \mathbf{i} \;\Big|\; \neg F \;\Big|\; F \wedge F \;\Big|\; F \vee F \;\Big|\; \mathbf{C}^q F \;\Big|\; \mathbf{D}^q F,$$

where $i \in \mathbb{N}$ and $q \in \mathbb{Q} \cap [0,1]$.

The definition of the semantics of $\mathbf{CPL}_0$ relies on the standard cylinder space $\big(2^{\mathbb{N}}, \sigma(\mathscr{C}), \mu_{\mathscr{C}}\big)$.

Definition 4 (Semantics of $\mathbf{CPL}_0$)**.** For each formula F of $\mathbf{CPL}_0$ its *interpretation* $[\![F]\!] \in \mathscr{B}(2^{\mathbb{N}})$ is the measurable set defined as follows:

$$\begin{aligned}
[\![\mathbf{i}]\!] &:= Cyl(i) \\
[\![\neg G]\!] &:= 2^{\mathbb{N}} - [\![G]\!] \\
[\![G \wedge H]\!] &:= [\![G]\!] \cap [\![H]\!] \\
[\![G \vee H]\!] &:= [\![G]\!] \cup [\![H]\!]
\end{aligned}
\qquad\qquad
\begin{aligned}
[\![\mathbf{C}^q G]\!] &:= \begin{cases} 2^{\mathbb{N}} & \text{if } \mu_{\mathscr{C}}\big([\![G]\!]\big) \geq q \\ \emptyset & \text{otherwise} \end{cases} \\
[\![\mathbf{D}^q G]\!] &:= \begin{cases} 2^{\mathbb{N}} & \text{if } \mu_{\mathscr{C}}\big([\![G]\!]\big) < q \\ \emptyset & \text{otherwise.} \end{cases}
\end{aligned}$$

Example 1. Let us consider $\mathbf{C}^{1/2}(F \vee G)$, where $F = \mathbf{0} \wedge \neg\mathbf{1}$ and $G = \neg\mathbf{0} \wedge \mathbf{1}$. The measurable sets, $[\![F]\!]$ and $[\![G]\!]$, have both measure $\frac{1}{4}$ and are disjoint. Hence, $\mu_{\mathscr{C}}\big([\![F \vee G]\!]\big) = \mu_{\mathscr{C}}\big([\![F]\!]\big) + \mu_{\mathscr{C}}\big([\![G]\!]\big) = \frac{1}{2}$, and $[\![\mathbf{C}^{1/2}(F \vee G)]\!] = 2^{\mathbb{N}}$.

Observe that counting quantifiers are inter-definable but not dual, in the sense of standard modal operators: $\mathbf{C}^q F$ is *not* equivalent to $\neg\mathbf{D}^q\neg F$.

Definition 5. For any formula of $\mathbf{CPL}_0$, call it F, F is said to be *valid* when $[\![F]\!] = 2^{\mathbb{N}}$ and to be *invalid* when $[\![F]\!] = \emptyset$.

Remark 2. By combining counting quantifiers it is possible to easily express that a formula F "is true with precisely a given probability", see [29].

B.2.2. Syntax and Semantics of CPL

We now introduce the more expressive fragment **CPL**, in which relations between valuations of different groups of variables can be considered. Its language is made of *named* atomic formulas and *named* counting quantifiers. The corresponding quantitative semantics is subtler than the one for $\mathbf{CPL}_0$ and, in particular, the interpretation of counting-quantified formulas relies on some technical notions.

Notation 2. We use $a, b, c... \in \mathscr{A}$ for names and $X, Y... \subseteq \mathscr{A}$ for (countable) sets of names.

Intuitively, named counting quantifiers, $\mathbf{C}^q_a$ or $\mathbf{D}^q_a$, count the number of valuations of propositional atoms *with the corresponding name*, here a, satisfying the argument formula.

Definition 6 (Formulas of **CPL**)**.** *Formulas of **CPL*** are defined by the grammar below:

$$F ::= \mathbf{i}_a \;\Big|\; \neg F \;\Big|\; F \wedge F \;\Big|\; F \vee F \;\Big|\; \mathbf{C}^q_a F \;\Big|\; \mathbf{D}^q_a F,$$

where $i \in \mathbb{N}$, $a \in \mathscr{A}$, and $q \in \mathbb{Q} \cap [0, 1]$.

A named quantifier binds the occurrences of the name in the argument formula and counts models *relative* to the corresponding bounded variable. The intuitive meaning of $\mathbf{C}^q_a F$ is that F is true in at least q valuations of the variables with name a.

The interpretation of a formula F now depends on the choice of a finite set of names $X \supseteq \mathrm{FN}(F)$ and is a measurable set $[\![F]\!]_X$ belonging to the Borel algebra, $\mathscr{B}\big((2^{\mathbb{N}})^X\big)$. To define it formally we need to introduce the technical notion of f-projection.

Definition 7 (f-projection)**.** Let X, Y be two disjoint, finite sets of names, and $f \in (2^{\mathbb{N}})^X$. For all $\mathcal{X} \subseteq (2^{\mathbb{N}})^{X \cup Y}$, the *$f$-projection* of $\mathcal{X}$ is the set:

$$\Pi_f(\mathcal{X}) := \big\{g \in (2^{\mathbb{N}})^Y \mid f + g \in \mathcal{X}\big\} \subseteq (2^{\mathbb{N}})^Y,$$

where

$$(f + g)(\alpha) := \begin{cases} f(\alpha) & \text{if } \alpha \in X \\ g(\alpha) & \text{if } \alpha \in Y. \end{cases}$$

Definition 8 (Semantics of **CPL**)**.** For each formula F of **CPL** and finite set of names such that $X \supseteq \mathrm{FN}(F)$, the *interpretation* $[\![F]\!]_X \subseteq (2^{\mathbb{N}})^X$ is defined as follows:

$$\begin{aligned}
[\![\mathbf{i}_a]\!]_X &:= \{f \mid f(a)(i) = 1\} & [\![\neg G]\!]_X &:= (2^{\mathbb{N}})^X - [\![G]\!]_X \\
[\![G \wedge H]\!]_X &:= [\![G]\!]_X \cap [\![H]\!]_X & [\![\mathbf{C}^q_a G]\!]_X &:= \big\{f \mid \mu_{\mathscr{C}}\big(\Pi_f([\![F]\!]_{X \cup \{a\}})\big) \geq q\big\} \\
[\![G \vee H]\!]_X &:= [\![G]\!]_X \cup [\![H]\!]_X & [\![\mathbf{D}^q_a G]\!]_X &:= \big\{f \mid \mu_{\mathscr{C}}\big([\![G]\!]_{X \cup \{a\}}\big) < q\big\}.
\end{aligned}$$

Example 2. Let us consider the formula $\mathbf{C}^{1/2}_b \mathbf{C}^{1/2}_a F$, where

$$F = \big(\mathbf{2}_a \wedge (\neg \mathbf{2}_b \wedge \mathbf{3}_b)\big) \vee \big(\neg \mathbf{2}_a \wedge (\mathbf{2}_b \wedge \neg \mathbf{3}_b)\big) \vee \big((\neg \mathbf{2}_a \wedge \mathbf{3}_a) \wedge \mathbf{3}_b\big).$$

The valuations $f \in (2^{\mathbb{N}})^{\{b\}}$ belonging to $[\![\mathbf{C}^{1/2}_a F]\!]_{\{b\}}$ are those which can be extended to valuations of all Boolean variables satisfying F in at least half of the cases. Let us list all possible cases:

1. $f(b)(2) = f(b)(3) = 1$. Then, F has $\frac{1}{4}$ chances of being true, as both $\neg \mathbf{2}_a$ and $\mathbf{3}_a$ have to be true
2. $f(b)(2) = 1$, $f(b)(3) = 0$. Then, F has $\frac{1}{2}$ chances of being true, as $\neg \mathbf{2}_a$ has to be true
3. $f(b)(2) = 0$ and $f(b)(3) = 1$. Then, F has $\frac{3}{4}$ chances of being true, as either $\mathbf{2}_a$ or both $\neg \mathbf{2}_a$ and $\mathbf{3}_a$ have to be true
4. $f(b)(2) = f(b)(3) = 0$. Then, F has no chance of being true.

Clearly, $[\![\mathbf{C}^{1/2}_a F]\!]_{\{b\}}$ only contains the valuations which agree with cases 2. and 3. Therefore, $\big[\!\big[\mathbf{C}^{1/2}_b \mathbf{C}^{1/2}_a F\big]\!\big]_{\emptyset} = 2^{\mathbb{N}}$, i.e. $\mathbf{C}^{1/2}_b \mathbf{C}^{1/2}_a F$ is valid, since half of the valuations of b has at least $\frac{1}{2}$ chances of being extended to a model of F.

B.3. On Measure-Quantified Peano Arithmetic

The standard model $\mathcal{N} = (\mathbb{N}, +, \times)$ has nothing probabilistic in itself. To obtain a model for **MQPA** we extend it to a probability space, obtaining $\mathcal{P} = (\mathbb{N}, +, \times, \sigma(\mathscr{C}), \mu_{\mathscr{C}})$. The grammar for terms of **MQPA** is standard, whereas the syntax for formulas is obtained by endowing the language of **PA** with special *flipcoin formulas* of the form $\textsc{Flip}(t)$ and *measure-quantified formulas*, such as $\mathbf{C}^{t/s}F$ and $\mathbf{D}^{t/s}F$. Specifically, $\textsc{Flip}(\cdot)$ is a special unary predicate with an intuitive computational meaning: it provides an infinite sequence of independently and randomly distributed bits. Given a closed term t, $\textsc{Flip}(t)$ holds if and only if the n-th tossing returns 1, where n denotes $t + 1$.

Definition 9 (Terms and Formulas of **MQPA**)**.** Let $\mathcal{G}$ be a denumerable set of *ground variables*, whose elements are indicated by metavariables such as $x, y, \ldots$. The *terms of **MQPA***, denoted by $t, s, \ldots$, are defined by the grammar below:

$$t ::= x \;\Big|\; \mathtt{0} \;\Big|\; \mathtt{S}(t) \;\Big|\; t + s \;\Big|\; t \times s.$$

The syntax for *formulas of **MQPA*** is as follows:

$$F ::= \textsc{Flip}(t) \;\Big|\; (t = s) \;\Big|\; \neg F \;\Big|\; F \vee G \;\Big|\; F \wedge G \;\Big|\; \exists x.F \;\Big|\; \forall x.F \;\Big|\; \mathbf{C}^{t/s}F \;\Big|\; \mathbf{D}^{t/s}F.$$

Given an environment $\xi : \mathcal{G} \to \mathbb{N}$, the interpretation $[\![t]\!]_\xi$ of a term t is defined as usual. Instead, the interpretation of formulas requires a little care, being it inherently *quantitative*; indeed, any formula of **MQPA**, say F, is associated with a *measurable* set, $[\![F]\!]_\xi \in \sigma(\mathscr{C})$.

Definition 10 (Semantics for Formulas of **MQPA**)**.** Given a formula F and an environment ξ, the *interpretation of F in ξ*, $[\![F]\!]_\xi \in \sigma(\mathscr{C})$, is the measurable set of sequences inductively defined as follows:

$$[\![\textsc{Flip}(t)]\!]_\xi := \{\omega \mid \omega([\![t]\!]_\xi) = 1\}$$

$$[\![(t = s)]\!]_\xi := \begin{cases} 2^{\mathbb{N}} & \text{if } [\![t]\!]_\xi = [\![s]\!]_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

$$[\![\neg G]\!]_\xi := 2^{\mathbb{N}} - [\![G]\!]_\xi$$

$$[\![G \vee H]\!]_\xi := [\![G]\!]_\xi \cup [\![H]\!]_\xi$$

$$[\![G \wedge H]\!]_\xi := [\![G]\!]_\xi \cap [\![H]\!]_\xi$$

$$[\![\exists x.G]\!]_\xi := \bigcup_{i \in \mathbb{N}} [\![G]\!]_{\xi\{x \leftarrow i\}}$$

$$[\![\forall x.G]\!]_\xi := \bigcap_{i \in \mathbb{N}} [\![G]\!]_{\xi\{x \leftarrow i\}}$$

$$[\![\mathbf{C}^{t/s}G]\!]_\xi := \begin{cases} 2^{\mathbb{N}} & \text{if } [\![s]\!]_\xi > 0 \text{ and } \mu_{\mathscr{C}}([\![G]\!]_\xi) \geq [\![t]\!]_\xi / [\![s]\!]_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

$$[\![\mathbf{D}^{t/s}G]\!]_\xi := \begin{cases} 2^{\mathbb{N}} & \text{if } [\![s]\!]_\xi = 0 \text{ or } \mu_{\mathscr{C}}([\![G]\!]_\xi) < [\![t]\!]_\xi / [\![s]\!]_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

The semantics is well-defined as the sets $[\![\textsc{Flip}(t)]\!]_\xi$ and $[\![(t = s)]\!]_\xi$ are measurable, and measurability is preserved by all logical and counting operators. A formula of **MQPA**, say F, is said to be *valid* when, for every ξ, $[\![F]\!]_\xi = 2^{\mathbb{N}}$.

Example 3. The formula $F = \mathbf{C}^{1/1}\exists x.\textsc{Flip}(x)$ states that a true random bit will almost surely be met. This formula is valid, as the set of constantly 0 sequences forms a singleton of measure 0.