KI - Künstliche Intelligenz
https://doi.org/10.1007/s13218-024-00843-y

DISSERTATION AND HABILITATION ABSTRACTS

Towards a Logical Foundation of Randomized Computation: Doctoral Thesis Abstract

Melissa Antonelli[1]

Received: 16 December 2023 / Accepted: 24 March 2024

Abstract
Interactions between logic and theoretical computer science are multiple and profound. In the last decades, they have been deeply investigated, but, surprisingly, the study of probabilistic computation was only marginally touched by such fruitful interchanges. The overall goal of my doctoral thesis was precisely that of start bridging this gap by developing logical systems corresponding to specific aspects of randomized computation and, due to them, 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.

Keywords Randomized computation · Logical foundations of computer science · Probability logic · Reasoning about uncertainty

1 Introduction

Among the features historically defining standard computational models there is certainly *determinism*: given an algorithm and input, the sequence of computation steps is uniquely determined. In 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.

This more flexible design makes probabilistic models very efficient and powerful tools [33], with several applications in computer science (CS) and technology. Generally speaking, these are often crucial when dealing with *uncertain* information or *partial* knowledge, namely for all systems acting in realistic contexts, think for examples of driverless cars [40] or of computer vision modelling [29]. Notably, in some areas probabilistic models have become even more than optional; for instance in cryptography, where secure encryption schemas are probabilistic [25].

1.1 The Dissertation

In this context, my Ph.D. dissertation was driven by two main considerations. On the one hand, since their appearance in the 1950s, probabilistic computational models have become ubiquitous in several fast-growing areas of CS, and, by now, related, abstract machines—as probabilistic Turing machines (PTMs) [19, 22, 36], stochastic automata [13, 18, 34] or randomized λ-calculi [28, 35]—have been massively studied in the literature. On the other, there exist deep and mutual interactions linking logic and theoretical computer science (TCS) and, in the past, the development of computational models and theory has considerably benefitted from them. Surprisingly, 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 probabilistic algorithms in many relevant fields of IT, from AI to statistical learning, from cryptography to approximate computing and robotics.

The global purpose of my doctoral thesis consisted in laying the foundation for a uniform approach to bridge the

The author thanks Helsinki Institute for Information Technology (HIIT) for supporting her work since 2023.

✉ Melissa Antonelli
melissa.antonelli@helsinki.fi

[1] HIIT (Helsinki Institute for Information Technology), University of Helsinki, Pietari Kalmin katu, 5, 00560 Helsinki, Finland

Published online: 16 April 2024

mentioned gap. To do so, the key ingredient is the introduction of a family of new logics, whose language includes non-standard quantifiers "measuring" the probability for the corresponding argument formula to be true and associated with inherently quantitative semantics.[1]

Concretely, the dissertation is tripartite. The first part focusses on the relation between logic and counting complexity, and its main result consists in showing that classical counting propositional logic provides a purely logical characterization of Wagner's hierarchy [43]. The second part of the thesis deals with programming language theory. Here, the Curry–Howard correspondence (CHC) [38] is extended for the first time to the probabilistic setting by relating the intuitionistic version of our counting logic and a counting-typed probabilistic λ-calculus. Finally, we consider the link between arithmetic and computation by introducing a quantitative extension of the language of Peano arithmetic (**PA**) able to formalize basic results from probability theory. This language is also our starting point to define *randomized* bounded arithmetic and to generalize canonical results by Buss [12].

2 Relating Logic and Randomized Computation

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, defining 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—such as programming language theory [38], verification [39] and database theory [15], computational and descriptive complexity [16], just to name a few - have effectively taken advantages from this mutual dialogue. As Siekmann wrote, "[i]n many respects, logic provides computer science with both a unifying foundational framework and a tool for modeling" [37, 14, 16, 27, 41], and by the numerous *concrete* exchanges between these disciplines: while the growing importance of IT has guided and stimulated advances in logic, logical tools have extensive applications in CS and technology.

On the other hand, when switching to the randomized setting, such a deep correspondence has been investigated only sparsely. One crucial peculiarity of dealing with probabilistic algorithms is that, in this case, behavioral properties, like termination or equivalence, have an *inherently quantitative* nature, that is a computation terminates *with (at least or at most) a given probability*, and a program might simulate a 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? My Ph.D. dissertation offers a positive answer at least to the specific aspects of the interaction between quantitative logics and randomized computation it focusses on.

2.1 Counting 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 **NP**-complete problem [16], while the language of classical tautologies is **coNP**-complete. In the early 1970s, Meyer and Stockmeyer also showed that, when switching to *quantified* propositional logic (QPL), the full polynomial hierarchy 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 [31, 32]. 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, introduced by Valiant [42] and Wagner [43]:

$$\text{polynomial hierarchy} : \mathsf{QPL} \Leftrightarrow \text{counting hierarchy} : ?$$

In the first part of the dissertation, a counting propositional system, called CPL, is introduced. This logic is a generalization of **PL** able to express that a formula is true *with (at least or at most) a given probability* [1, 9]. CPL is shown to be strongly related to counting computation and classes, being the probabilistic counterpart of QPL [6, 9, 10]. Indeed, its counting quantifiers can be naturally seen as "quantitative" versions of standard propositional ones, and our main result here is the *purely logical* characterization of Wagner's hierarchy via complete problems defined in terms of CPL-formulas.

2.2 Programming Language Theory

Traditionally, CHC relates intuitionistic **PL** and the simply-typed λ-calculus [38], but in the last fifty years it was shown to hold in other, more sophisticated contexts too. Meanwhile, randomized λ-calculi [35] and associated type systems, sometimes also guaranteeing desirable forms of termination [20], were introduced. Yet, these were not designed having a logical system in mind, and no (probabilistic) CHC is known for them:

[1] An intuitive presentation of counting logics can be found in Sect. 3. For further details, see [6–8].

simply typed $\lambda_{\rightarrow}$: iPL $\Leftrightarrow$ randomized λ-calculi : ?

In the second part of the thesis, two new systems are introduced to define the first probabilistic version of the above correspondence. On the one hand, we consider the intuitionistic counterpart of univariate CPL, called iCPL_0, and show it able to capture quantitative behavioral properties. On the other, we define a "counting-typed" probabilistic λ-calculus. Its untyped part is strongly inspired by the probabilistic event λ-calculus presented in [17], while the type system is defined mimicking counting quantifiers. Finally, we establish a (static and dynamic) correspondence, in the style of Curry and Howard, between these two systems [8, 10].

2.3 Probabilistic (Bounded) Arithmetic

2.3.1 Arithmetic and Computation Theory

The theory of (deterministic) computation and arithmetic are linked by deep results coming from logic and recursion theory, such as Gödel's arithmetization [23], or realizability [30], or the *Dialectica* interpretation [24]. Many interesting properties of algorithms can be expressed in the arithmetical language, 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 considering the probabilistic realm, there is no theory relating to randomized computation as **PA** does to deterministic one:

det. comput. : **PA** $\Leftrightarrow$ prob. comput. : ?

In the third part of the dissertation, we present a quantitative extension of the language of arithmetic, 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. This language is also proved to be actually connected to randomized computation as we establish the probabilistic version of Gödel's arithmetization [17], namely it is shown that any *random* function can be expressed by a formula of MQPA.

2.3.2 Bounded Arithmetic and Probabilistic Complexity

In addition, the language of MQPA is at the basis of our study of randomized bounded arithmetic theories. Historically, one of the main motivations for the development of bounded arithmetics (i.e. subsystems of **PA** whose language includes symbols for functions with specific growth rate together with bounded quantifiers, and in which induction is variously limited) was their connection with computational complexity [12]. As it is clear that not all computable functions are *feasibly* computable, bounded theories have become essential to characterize interesting (feasible) complexity classes in terms of families of arithmetic formulas. Specifically, in 1986, Buss proved that the class of poly-time computable functions precisely corresponds to that of functions which are Σ^b_1-definable in a given bounded theory, S^1_2. Although this fact is very insightful, no similar result was established in the probabilistic framework:

deterministic classes : BA $\Leftrightarrow$ probabilistic classes : ?

Inspired by MQPA, in the third part of the thesis we introduce a *randomized* bounded theory, called RS^1_2, enabling us to logically capture relevant probabilistic classes, as **BPP** [2] [3–5].

3 From Evaluating to Measuring

Counting quantifiers are quantifiers of the form $\mathbf{C}^q$ or $\mathbf{D}^q$ (for $q \in \mathbb{Q} \cap [0,1]$) and capable of expressing probabilities within a logical language. Intuitively, a counting quantified formula $\mathbf{C}^q F$ expresses that F is true with probability greater than or equal to q, while $\mathbf{D}^q F$ expresses that F has probability strictly smaller than q of being true. Thus, these quantifiers not only determine the *existence* of a satisfying assignment, but also count *how many* those assignments are. In a sense, they are *quantitative* generalizations of standard propositional ones. Accordingly, we move from a standard language made of formulas of the form $(\forall X)F, (\exists X)F$ to that of counting quantified ones, $\mathbf{C}^q F, \mathbf{D}^q F$.

Such a generalization is possible only when contextually switching from a truth-functional (i.e. $[\![F]\!]_{\mathsf{QPL}} \in \{0,1\}$) to a quantitative semantics, in which formulas are no more interpreted as single truth-values but as measurable sets of models (i.e. $[\![F]\!]_{\mathsf{CPL}_0} \subseteq 2^{\mathbb{N}}$). 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.

Example 1 The (pseudo-)counting formula $\mathbf{C}^{1/4}(X \wedge Y)$ 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.

[2] **BPP** is the class of decision problems solvable by a *poly-time* PTM with error probability smaller than $\frac{1}{3}$ *for any instance*. Differently from **P** it is a *semantic* class, the definition of which relies on algorithms to be both efficient and not *too erratic*. For further discussion, see [5, pp. 1–5].

In this way, such logic allows 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 [1], but, as predictable, their expressive power is quite limited. So, as anticipated, we have generalized the notion of counting quantifier to define 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.

3.1 On Counting Propositional Logic

In order to make these intuitive notions clearer we briefly introduce the univariate fragment CPL_0. Although the expressive power of this logic is limited, its semantics has a very natural interpretation and can be extended to full CPL in a straightforward way.

When dealing with 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" (and belonging to the standard Borel algebra over $2^{\mathbb{N}}$, $\mathscr{B}(2^{\mathbb{N}})$). In particular, atomic propositions are interpreted as special cylinder sets [11] 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 $\mathbf{C}^q$ and $\mathbf{D}^q$ for $q \in \mathbb{Q} \cap [0,1]$.[3] As seen, the formula $\mathbf{C}^q F$ (resp., $\mathbf{D}^q F$) intuitively expresses that F is satisfied by a portion of assignments greater (resp., strictly smaller) than q. For example, $\mathbf{C}^{1/2} F$ expresses that F is satisfied by *at least* half of its valuations.

A bit more formally,

Definition 1 (Formulas of CPL_0) *Formulas of* CPL_0 are defined by the grammar below,

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

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

[3] Notice that our quantifiers have been inspired by Wagner's counting operator [43].

The definition of the semantics for CPL_0 relies on the standard cylinder space $\left(2^{\mathbb{N}}, \sigma(\mathscr{C}), \mu_{\mathscr{C}}\right)$.[4] x

Definition 2 (Semantics of CPL_0) For each formula F of CPL_0 its *interpretation*, $[\![F]\!] \in \mathscr{B}(2^{\mathbb{N}})$, is the measurable set:

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

Example 2 Let $\mathbf{C}^{1/2}(F \vee G)$, 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, $\mu_{\mathscr{C}}([\![F \vee G]\!]) = \mu_{\mathscr{C}}([\![F]\!]) + \mu_{\mathscr{C}}([\![G]\!]) = \frac{1}{2}$ and $[\![\mathbf{C}^{1/2}(F \vee G)]\!] = 2^{\mathbb{N}}$.

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

In more expressive CPL, relations between valuations of different groups of variables can be taken into account. Contextually, the corresponding quantitative semantics is subtler than that of CPL_0, and to define the interpretation for counting quantified formulas we rely on a few technical notions.[5] Remarkably, there is a strong connection between (closed) formulas of CPL_0 and (closed) formulas of CPL in which only one name occurs.[6] Moreover, in [6, 9, 10], sound and complete proof system(s) for CPL_0 and CPL have also been introduced.

3.2 On Measure-Quantified Peano Arithmetic

The standard model $\mathscr{N} = (\mathbb{N}, +, \times)$ has nothing probabilistic in itself. So, to define a model for MQPA we extend it to a probability space, obtaining $\mathscr{P} = (\mathbb{N}, +, \times, \sigma(\mathscr{C}), \mu_{\mathscr{C}})$. The grammar for terms of MQPA is standard, while that for formulas is obtained by endowing the language of **PA** with special *flipcoin formulas* of the form $\textsc{Flip}(t)$ and *measure-quantified formulas*, namely, $\mathbf{C}^{t/s} F$ and $\mathbf{D}^{t/s} F$ (where, now,

[4] Here, $\mathscr{C}$ is the field set of all cylinders of any rank, $\sigma(\mathscr{C})$ is the σ-algebra generated by $\mathscr{C}$, and $\mu_{\mathscr{C}}$ denotes the standard cylinder measure over $\sigma(\mathscr{C})$, i.e. the unique measure on $\sigma(\mathscr{C})$ such that, for any $i \in \mathbb{N}$, $\mu_{\mathscr{C}}(Cyl(i)) = \frac{1}{2}$. See [11].

[5] For further details, see [6, 9, 10].

[6] In [2, Sec. 4.1], a straightforward, validity-preserving translation from (closed) formulas of CPL_0 to (closed) formulas of CPL and vice-versa is presented.

t and s are terms, possibly including variables). Specifically, $\textsc{Flip}(\cdot)$ is a special unary predicate with an intuitive computational meaning: it provides an infinite sequence of independently and identically distributed bits. Given a closed term t, $\textsc{Flip}(t)$ holds when the n-th tossing returns 1, and n is $t + 1$.

Definition 3 (Terms and Formulas of MQPA) Let $\mathcal{G}$ be a denumerable set of *ground variables*, whose elements are indicated by metavariables $x, y, \ldots$. The *terms of* MQPA, denoted by $t, s, \ldots$, are defined by the grammar below:

$$t ::= x \mid 0 \mid \mathsf{S}(t) \mid t + s \mid t \times s.$$

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

$$F ::= \textsc{Flip}(t) \mid t = s \mid \neg F \mid F * G \mid \exists x.F \mid \forall x.F \mid \blacksquare F,$$

for $* \in \{\vee, \wedge\}$ and $\blacksquare \in \{\mathbf{C}^{t/s}, \mathbf{D}^{t/s}\}$.

Given an environment $\xi : \mathcal{G} \to \mathbb{N}$, the interpretation for a term t, $[\![t]\!]_\xi$, is defined as usual. Instead, that of formulas is not, being it inherently *quantitative*.

Definition 4 (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:

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

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

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

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

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

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

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

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

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

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. The formula is valid as the set of constantly 0 sequences forms a singleton of measure 0.

4 Conclusion and Future Work

My Ph.D. thesis aims at being a first step to bridge logic and probabilistic computation. In it quantitative logical systems are developed to *uniformly* generalize standard achievements in TCS to the probabilistic setting. First, classical CPL_0 and CPL are introduced and proved to be strongly connected to counting classes, as formulas of CPL in a special prenex normal form provide complete problems for each level of Wagner's hierarchy [6, Cor. 1].[7] Then, the computational fragment of its intuitionistic version, iCPL_0, and the probabilistic CHC are defined: proofs in iCPL_0 correspond, in the sense of Curry and Howard, to typing derivations for the randomized λ-calculus Λ_{PE}, so that counting quantifiers "reveal" the probability of termination of the underlying probabilistic program [8, Sec. 5].[8] In addition, a quantitative extension of the language of **PA**, able to formalize basic results from probability theory, which are not expressible in standard arithmetic, is presented together with the first randomized version of Gödel's arithmetization [7, Th. 3].[9] Finally, a randomized bounded theory *á la Buss* is defined such that bounded formulas provably total in it precisely capture poly-time random functions. Due to $\mathbf{RS}^1_2$, a new, syntactical characterization of **BPP** is obtained by internalizing the error-bound check within the logical system [5, Th. 15, 18].

To the best of my knowledge, the project and approach developed in the dissertation is quite new. Accordingly, several problems and challenges are still open. In general, the investigation of the expressive power of our logics (initiated in [1, Sect. 3]) and of their relation with probability and modal systems deserves further attention. About the proof theory of CPL_0 and CPL, the study of their dynamic (namely, the underlying cut-elimination procedure) and the introduction of a purely syntactical calculus have only been initiated.[10] Also the introduction of intuitionistic logics and probabilistic CHC opens up several new avenues of research: from the extension of CHC to polymorphic types

[7] A bit more formally, the theorem states that closed and valid CPL -formulas in PPNF and with k-ary quantifier alterations define a complete set for $\mathbf{CH}_k$, where CPL is the multivariate version of CPL_0, and a formula of CPL is in PPNF if it is both in PNF and **D**-free.

[8] Extending the system with intersection types leads to a full characterization of termination probability, see [8, Sec. 6].

[9] The theorem states that all computable random functions are arithmetical, where a random function $f : \mathbb{N}^m \to \mathcal{D}_{\mathbb{N}}$ is said to be arithmetical if there is a formula of MQPA, F_f, with free variables $x_1, \ldots, x_m, y$ such that for every $n_1, \ldots, n_m, l \in \mathbb{N}$, $\mu_{\mathscr{C}}([\![F_f(n_1, \ldots, n_m, l)]\!]) = f(n_1, \ldots, n_m)(l)$, see [7, Df. 4].

[10] The calculi introduced in [6, 9] include external hypotheses, corresponding to oracle queries counting the number of satisfying Boolean assignments. For first ideas to design a calculus without appealing for an external source, see [1].

or to control operator to the study of intersection types to support program synthesis.

Concerning measure-quantified languages of arithmetic, one of the most compelling problems is the definition of a corresponding sound and sufficiently expressive proof system. Furthermore, as the language of MQPA is somehow minimal "by design", it would be natural to generalize its study to more expressive (named) fragments, following the path delineated by multivariate CPL. At the same time, the introduction of randomized bounded arithmetic could be the starting point for a long-term study on the logical nature of semantic classes and on its link with proof complexity, think for example of natural extensions of our approach to the characterization of other randomized classes, such as **ZPP**, **RP** and **coRP**, or to its applications to the study of random resolution refutations.

Acknowledgements I thank my Ph.D. thesis supervisor U. Dal Lago and co-supervisor P. Pistone for their constant guidance and crucial help: all the results presented in my dissertation are part of a joint work with them. I am grateful to the anonymous reviewers for their helpful comments.

Funding Open Access funding provided by University of Helsinki (including Helsinki University Central Hospital).

Data availability The data and proofs generated during the current study are available from the corresponding author on reasonable request.

References

1. Antonelli M (2022) Two remarks on counting propositional logic. In: Proceedings of the BEWARE, pp 20–32
2. Antonelli M (2023) Towards a logical foundation of randomized computation. Ph.D. thesis, Department of Computer Science and Engineering, University of Bologna
3. Antonelli M, Dal Lago U, Davoli D, Oitavem I, Pistone P (2022) Towards a randomized bounded arithmetic. In: AILA—book of abstract
4. Antonelli M, Dal Lago U, Davoli D, Oitavem I, Pistone P (2023) Enumerating error bounded poly-time algorithms through arithmetical theories. In: Logic colloquium 2023—book of abstract, pp 45–46
5. Antonelli M, Dal Lago U, Davoli D, Oitavem I, Pistone P (2024) Enumerating error bounded poly-time algorithms through arithmetical theories. In: CSL 2024
6. Antonelli M, Dal Lago U, Pistone P (2021) On counting propositional logic and Wagner's hierarchy. In: Proceedings of the ICTCS, pp 107–121
7. Antonelli M, Dal Lago U, Pistone P (2021) On measure quantifiers in first-order arithmetic In: Proceedings of the CiE, pp 12–24
8. Antonelli M, Dal Lago U, Pistone P (2022) Curry and Howard meet borel. In: Proceedings of the LICS, pp 1–13
9. Antonelli M, Dal Lago U, Pistone P (2023) On counting propositional logic and Wagner's hierarchy. Theor Comput Sci 966–967
10. Antonelli M, Dal Lago U, Pistone P (2023) Towards logical foundations for probabilistic computation. APAL (in press)
11. Billingsley P (1995) Probability and measure. Wiley, New York
12. Buss S (1896) Bounded arithmetic. Ph.D. thesis, Princeton University
13. Carlyle JM (1963) Reduced forms for stochastic sequential machines. J Math Anal 7:167–174
14. Church A, Kleene S (1936) Formal definitions in the theory of ordinal numbers. Fund Math 28:11–21
15. Codd E (1972) Relational completeness of data base sublanguages. In: Proceedings of the 6th courant computer science symposium, pp 65–98
16. Cook S (1971) The complexity of theorem-proving procedures. In: Proceedings of the STOC, pp 151–158
17. Dal Lago U, Guerrieri G, Heijltjes W (2020) Decomposing probabilistic lambda-calculi. In: Proceedings of the FoSSaCS, pp 136–156
18. Davis AS (1961) Markov chains as random input automata. Am Math Mon 68(3):264–267
19. De Leeuw K et al (1956) Computability by probabilistic machines. In: Automata studies, vol 34, pp 91–95
20. Faggian C, Ronchi della Rocca S (2019) Lambda calculus and probabilisitic computation. In: Proceedings of the LICS, pp 1–13
21. Fagin R, Halpern J, Megiddo N (1990) A logic for reasoning about probabilities. Inf Comput 87:78–128
22. Gill J (1974) Computational complexity of probabilistic Turing machine. In: Proceedings of STOC, pp 91–95
23. Gödel K (1931) Über formal unentscheidbare Sätze der principia mathematica and Verwandter systeme. Monatsch Math Phys 38:171–178
24. Gödel K (1958) Über eine Bisher noch nicht Benützte erweiterung des finiten standdpunktes. Dialectica 12:280–287
25. Goldwasser S, Micali S (1984) Probabilistic encryption. J Comput Syst Sci 28:279–299
26. Halpern J (2003) Reasoning about uncertainty. MIT Press, New York
27. Howard W (1980) The formulae-as-types notion of construction. In: Seldin J, Hindley J (eds) To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism. Academic Press, London, pp 479–490
28. Jones C, Plotkin G (1989) A probabilistic power domain for evaluations. In: Proceedings of the LICS, pp 186–195
29. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press, New York
30. Kreisel G (1959) Interpretation of analysis by means of constructive functionals of finite types. In: Heyting A (ed) Constructivity in mathematics. North-Holland, pp 101–128
31. Meyer AR, Stockmeyer LJ (1973) Word problems requiring exponential time (preliminary report). In: Proceedings of the STOC, pp 1–9
32. Meyer AR, tockmeyer LJ (1972) The equivalence problem for regular expressions with squaring requires exponential space. In: Proceedings of the SWAT, pp 125–129
33. Motwani R, Raghavan P (1995) Randomized algorithms. Cambridge University Press, Cambridge
34. Rabin MO (1963) Probabilistic automata. Inf Comput 6:230–245

35. Saheb-Djaromi N (1978) Probabilistic LCF. In: Press A (ed) Proceedings of the MFCS, pp 154–165
36. Santos E (1969) Probabilistic turing machines and computability. In: Proceedings of the AMS, vol 22, pp 704–710
37. Siekman J (2014) Computational logic. In: Siekmann J (ed) Handbook of the history of logic: computational logic, vol 9. Elsevier, London, pp 15–30
38. Sorensen M, Urzyczyn P (2006) Lectures on the Curry–Howard isomorphism, vol 149. Elsevier, New York
39. Thornton M, Drechsler R, Miller D (2001) Logic verification. Springer, New York, pp 201–230
40. Thrun S, Burgard W, Fox D (2006) Probabilistic robotics. MIT Press, New York
41. Turing A (1936) On computable numbers, with an application to the *Entscheidungsproblem*. In: Proceedings of the LMS, vol 42, pp 230–265
42. Valiant L (1979) The complexity of computing the permanent. TCS 8:189–201
43. Wagner K (1986) The complexity of combinatorial problems with succinct input representation. Acta Inform 23:325–356