On Counting Propositional Logic and Wagner's Hierarchy⋆ ⋆⋆

Melissa Antonelli, Ugo Dal Lago, and Paolo Pistone

Università di Bologna
{melissa.antonelli2,ugo.dallago,paolo.pistone2}@unibo.it

Abstract. We introduce and study counting propositional logic, an extension of propositional logic with counting quantifiers. This new kind of quantification makes it possible to express that the argument formula is true *in a certain portion* of all possible interpretations. We show that this logic, beyond admitting a satisfactory proof-theoretical treatment, can be related to computational complexity: the complexity of the underlying decision problem perfectly matches the appropriate level of Wagner's counting hierarchy.

Keywords: Propositional Logic · Counting Hierarchy · Computational Complexity

1 Introduction

Among the many intriguing relationships existing between logic and computer science, we can certainly mention the ones connecting classical propositional logic (**PL**, for short), on the one hand, and computational complexity, the theory of programming languages, and several other branches of theoretical computer science, on the other. As it is well known, **PL** provided the first example of a nontrivial NP-complete problem [11]; moreover, formal systems for classical and intuitionistic propositional logic correspond to type systems for λ-calculi and related formalisms [16, 34]. These lines of research have further evolved in various directions, resulting in active sub-areas of computer science. In particular, variations of propositional logic have been put in relation with complexity classes other than P and NP, as well as with type systems other than the simply typed λ-calculus. For instance, the complexity of deciding *quantified* propositional formulas is well known to match the appropriate level of the polynomial hierarchy (PH, for short) [27, 28, 35, 42, 10].

Nevertheless, some aspects of the theory of computation have not found a precise logical counterpart, at least so far. One such development concerns the

⋆⋆ Supported by ERC CoG "DIAPASoN", GA 818616 and ANR PRC project "PPS", ANR-19-CE48-0014.

counting classes of complexity and the related *counting hierarchy* (CH, for short), as introduced by Valiant [38] and Wagner [39–41], which are deeply connected to randomized complexity classes, such as PP. In fact, Wagner's CH has been treated logically by means of tools from descriptive complexity and finite model theory [22]. However, to the best of the authors' knowledge, there is no counterpart of counting classes in the realm of propositional logic.

In this paper we aim at bridging the gap by introducing a new quantified propositional logic, called counting propositional logic (**CPL**, for short). To present this logic in a more intuitive way, we start by defining a *univariate* fragment of **CPL**, that we call $\mathbf{CPL}_0$, and we later describe the general, *multivariate*, logic **CPL**. The main feature of both these logics is the presence of counting quantifiers, which are designed to count the number of values of the bound propositional variables satisfying the argument formula. We study the proof theory of counting logics together with its relations to computational complexity. Along the way, we introduce a sound and complete proof system in the form of a single-sided sequent calculus on labelled formulas. We also establish complexity results for both univariate $\mathbf{CPL}_0$, the validity of which corresponds to $\mathsf{P}^{\sharp\mathtt{SAT}}$, and for multivariate **CPL**, whose decision problem characterizes the whole CH. Indeed, we prove that deciding (a special kind of) prenex normal forms is complete for the appropriate level of the hierarchy, in the spirit of the correspondence between quantified propositional logic (**QPL**, for short) and PH.

The presentation is structured as follows. First, we introduce the syntax, semantics, and proof theory of counting logics. Specifically, in Section 2 we present a sound and complete proof system for $\mathbf{CPL}_0$, in the form of a labelled sequent calculus. In the univariate case, the correspondence with computational complexity is limited to the class $\mathsf{P}^{\sharp\mathtt{SAT}}$. In Section 3, we extend the calculus for $\mathbf{CPL}_0$ to the multivariate counting logic **CPL**. Section 4 is devoted to establishing the connection between counting logic and complexity theory, by relating the decision problem for **CPL** with the hierarchy CH. The proof proceeds by a careful analysis of prenex normal forms, which by construction have precisely the shape one needs to match Wagner's complete problems [40].

2 On Univariate Counting Propositional Logic

In this section we introduce a univariate version of counting propositional logic, called $\mathbf{CPL}_0$, together with a sound and complete proof system for it. Although this fragment has a limited expressive power, it provides an intuitive overview over the main semantical and proof-theoretical ingredients behind the more general logic **CPL**, introduced in the next section. Furthermore, the problem of establishing the validity of a $\mathbf{CPL}_0$-formula is proved to be in the class $\mathsf{P}^{\sharp\mathtt{SAT}}$.

2.1 $\mathbf{CPL}_0$-Formulae and their (Quantitative) Semantics

In the semantics of standard propositional logic the interpretation of a formula is a truth-value. The core idea from which our counting logics arise is to replace this way of interpreting formulas by a more *quantitative* semantics: the interpretation of a formula will be the *measurable set* of all valuations that satisfy it. Specifically, since propositional formulas may have an arbitrary number of propositional values, a valuation can be taken as an element of 2^ω; hence, given a formula of CPL_0, call it A, we may take as its interpretation the set $[\![A]\!] \subseteq 2^\omega$ made of all maps $f \in 2^\omega$ "making A true". Such sets can be easily seen to belong to the standard Borel algebra $\mathcal{B}(2^\omega)$ over 2^ω, thus yielding a genuinely quantitative semantics. In particular, atomic propositions are interpreted by *cylinder sets* [8] of the following form:

$$\mathrm{Cyl}(i) = \{f \in 2^\omega \mid f(i) = 1\}.$$

and non-atomic propositions are naturally interpreted by relying on the standard σ-algebra operations of complementation, finite intersection and finite union.

Since a formula corresponds to a measurable set, it makes sense to enrich the language of propositional logic with *new* formulas expressing conditions on the measure of such sets. By adapting Wagner's notion of counting operator [40, 41], we introduce two quantifiers, $\mathbf{C}^q, \mathbf{D}^q$, where q ranges over $\mathbb{Q}_{[0,1]}$, so that the formulas $\mathbf{C}^q A$ and $\mathbf{D}^q A$ express that A is satisfied by a given portion of all its possible interpretations. For example, the formula $\mathbf{C}^{\frac{1}{2}} A$ expresses the fact that A is satisfied by *at least half* of its valutations, namely A is true with probability at least $\frac{1}{2}$. Equally, the formula $\mathbf{D}^{\frac{3}{4}} A$ expresses the fact that A is satisfied by *strictly less than* three-fourths of its valutations, meaning that the probability for A to be true is strictly smaller than $\frac{3}{4}$. Semantically, this amounts at respectively checking that $\mu\big([\![A]\!]\big) \geq \frac{1}{2}$ and $\mu\big([\![A]\!]\big) < \frac{3}{4}$, where μ is the standard Borel measure on $\mathcal{B}(2^\omega)$.

Definition 1 (Formulas of CPL_0). *The* formulas of CPL_0 *are defined by the following grammar:*

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

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

In the following, let $\sigma(\mathscr{C})$ indicate the σ-algebra generated by the set of all n-cylinders, which is the smallest σ-algebra containing $\mathscr{C}$ and which is Borel. Moreover, let μ denote the standard cylinder measure over $\sigma(\mathscr{C})$, which can be defined as the unique measure on $\sigma(\mathscr{C})$ such that $\mu\big(\mathrm{Cyl}(i)\big) = \frac{1}{2}$, see [8].

Definition 2 (Semantics of CPL_0). *For each formula A of* CPL_0 *its* interpretation *is the measurable set, $[\![A]\!] \in \mathcal{B}(2^\omega)$, inductively defined as follows:*

$$\begin{aligned}
[\![\mathbf{i}]\!] &= \mathrm{Cyl}(i) \\
[\![\neg A]\!] &= 2^\omega \ - \ [\![A]\!] \\
[\![A \wedge B]\!] &= [\![A]\!] \cap [\![B]\!] \\
[\![A \vee B]\!] &= [\![A]\!] \cup [\![B]\!]
\end{aligned}
\qquad
\begin{aligned}
[\![\mathbf{C}^q A]\!] &= \begin{cases} 2^\omega & \text{if } \mu([\![A]\!]) \geq q \\ \emptyset & \text{otherwise} \end{cases} \\
[\![\mathbf{D}^q A]\!] &= \begin{cases} 2^\omega & \text{if } \mu([\![A]\!]) < q \\ \emptyset & \text{otherwise.} \end{cases}
\end{aligned}$$

Two CPL_0-formulas, A and B, are said *logically equivalent*, noted $A \equiv B$, when $[\![A]\!] = [\![B]\!]$. A formula A is *valid* when $[\![A]\!] = 2^\omega$.

The two counting quantifiers are inter-definable, as can be easily shown semantically:

$$\mathbf{C}^q A \equiv \neg \mathbf{D}^q A \qquad\qquad \mathbf{D}^q A \equiv \neg \mathbf{C}^q A. \tag{1}$$

Observe that they are not dual in the sense of standard modal operators: $\mathbf{C}^q A$ is *not* equivalent to $\neg \mathbf{D}^q \neg A$. Notably, using these quantifiers, it is even possible to express that a formula A is satisfied with probability *strictly greater than* q or with probability *no smaller than* q, as (resp.) $\mathbf{D}^{1-q} \neg A$ and $\mathbf{C}^{1-q} \neg A$. The example below can help clarifying the intuitive meaning of the semantics of CPL_0.

Example 1. Let us consider the counting formula $\mathbf{C}^{\frac{1}{2}} A$, where $A = B \vee C$, $B = \mathbf{0} \wedge \neg \mathbf{1}$ and $C = \neg \mathbf{0} \wedge \mathbf{1}$. The two measurable sets, $[\![B]\!]$ and $[\![C]\!]$, both have measure $\frac{1}{4}$ and are mutually disjoint. Hence, $\mu([\![A]\!]) = \mu([\![B]\!]) + \mu([\![C]\!]) = \frac{1}{2}$, which means that $[\![\mathbf{C}^{\frac{1}{2}} A]\!] = 2^\omega$, and so the formula $\mathbf{C}^{\frac{1}{2}} A$ is valid.

2.2 The Proof Theory of CPL_0

We introduce a one-sided, single-succedent sequent calculus, and prove that it is sound and complete with respect to the semantics of CPL_0. The language of this calculus is constituted by *labelled* formulas of the form $\mathscr{b} \rightarrowtail A$ or $\mathscr{b} \leftarrowtail A$, where A and $\mathscr{b}$ are respectively a counting and a Boolean formula. Intuitively, a labelled formula $\mathscr{b} \rightarrowtail A$ (resp. $\mathscr{b} \leftarrowtail A$) is true when the set of valuations satisfying $\mathscr{b}$ is included in (resp. includes) the interpretation of A.

Definition 3 (Boolean Formulas). Boolean formulas *are defined by the following grammar:*

$$\mathscr{b}, \mathscr{c} ::= \mathscr{x}_i \;\Big|\; \top \;\Big|\; \bot \;\Big|\; \neg \mathscr{b} \;\Big|\; \mathscr{b} \wedge \mathscr{c} \;\Big|\; \mathscr{b} \vee \mathscr{c}.$$

where $i \in \mathbb{N}$. *The* interpretation *of a Boolean formula* $\mathscr{b}$, $[\![\mathscr{b}]\!] \in \mathcal{B}(2^\omega)$, *is inductively defined as follows:*

$$[\![\mathscr{x}_i]\!] = \mathrm{Cyl}(i) \qquad\qquad [\![\neg \mathscr{b}]\!] = 2^\omega \quad [\![\mathscr{b}]\!]$$
$$[\![\top]\!] = 2^\omega \qquad\qquad [\![\mathscr{b} \wedge \mathscr{c}]\!] = [\![\mathscr{b}]\!] \cap [\![\mathscr{c}]\!]$$
$$[\![\bot]\!] = \emptyset \qquad\qquad [\![\mathscr{b} \vee \mathscr{c}]\!] = [\![\mathscr{b}]\!] \cup [\![\mathscr{c}]\!].$$

Labelled formulas are defined as follows:

Definition 4 (Labelled Formula). *A* labelled formula *is an expression of one of the forms* $\mathscr{b} \rightarrowtail A$, $\mathscr{b} \leftarrowtail A$, *where* $\mathscr{b}$ *is Boolean formula and* A *is a counting formula. A* labelled sequent *is a sequent of the form* $\vdash L$, *where* L *is a labelled formula.*

We also introduce a special class of formulas, that we call *external hypotheses*. Such formulas express semantic properties of Boolean formulas or conditions to be checked inside $\mathcal{B}(2^\omega)$. In the following, we use $\mathscr{b} \vDash \mathscr{c}$ as shorthand for $[\![\mathscr{b}]\!] \subseteq [\![\mathscr{c}]\!]$.

Definition 5 (External Hypothesis). *An* external hypothesis *is either an expression of the form* $\boldsymbol{b} \vDash \boldsymbol{c}$ *or of the form* $\mu([\![\boldsymbol{b}]\!]) \triangleright q$*, where* $\triangleright \in \{\geq, >, \leq, <, =\}$*,* $\boldsymbol{b}, \boldsymbol{c}$ *are Boolean formulas and* $q \in \mathbb{Q}_{[0,1]}$*.*

The measure of the interpretation of a Boolean formula, $\mu([\![\boldsymbol{b}]\!])$, can be related to the number $\sharp\mathtt{SAT}(\boldsymbol{b})$ of the valuations making $\boldsymbol{b}$ true as follows:

Lemma 1. *For each Boolean formula* $\boldsymbol{b}$ *containing the propositional variables* $\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n-1}$*,* $\mu([\![\boldsymbol{b}]\!]) = \sharp\mathtt{SAT}(\boldsymbol{b}) \cdot 2^{-n}$*.*

Proof. Any valuation $\theta : \{\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n-1}\} \to 2$ is associated to a measurable set $X(\theta) \in \mathcal{B}(2^\omega)$ by letting $X(\theta) = \{f \mid \forall_{i<n} f(i) = \theta(\boldsymbol{x}_i)\} = \bigcap_{i=0}^{n-1} \mathrm{Cyl}(i)^{\theta(\boldsymbol{x}_i)}$, where $\mathrm{Cyl}(i)^{\theta(\boldsymbol{x}_i)}$ is $\mathrm{Cyl}(i)$ if $\theta(\boldsymbol{x}_i) = 1$ and $\overline{\mathrm{Cyl}(i)}$ otherwise. Observe that $\mu(X(\theta)) = 2^{-n}$. For any $\boldsymbol{b}$, we have that $[\![\boldsymbol{b}]\!] = \bigcup_{\theta \vDash \boldsymbol{b}} X(\theta)$ (this is easily checked by induction on $\boldsymbol{b}$). Since for all distinct θ, θ', $X(\theta) \cap X(\theta') = \emptyset$, we conclude that $\sharp\mathtt{SAT}(\boldsymbol{b}) \cdot 2^{-n}$ $= \sum_{\theta \vDash \boldsymbol{b}} 2^{-n} = \sum_{\theta \vDash \boldsymbol{b}} \big(\mu(X(\theta))\big) = \mu\big(\bigcup_{\theta \vDash \boldsymbol{b}} X(\theta)\big) = \mu([\![\boldsymbol{b}]\!])$.

The calculus is defined by the rules in Figure 1. Let $\vdash_{\mathsf{CPL}_0} L$ indicate that $\vdash L$ is derivable by the given rules. In Figure 2 we provide an example of derivation in CPL_0.[1] The use of external hypotheses, that is, of genuinely semantic conditions, as premisses of syntactic rules might seem somehow unsatisfactory. However, such premisses do make sense from a computational viewpoint: they correspond to the idea that, when searching for a proof of a counting formula, one might need to call for an *oracle* for values of the form $\mu([\![\boldsymbol{b}]\!])$ (in fact, by Lemma 1, an oracle for $\sharp\mathtt{SAT}(\boldsymbol{b})$). This intuition will be made clear in the next subsection, where we discuss an algorithm for CPL_0-validity.

A labelled formula $\boldsymbol{b} \rightarrowtail A$ (resp. $\boldsymbol{b} \leftarrowtail A$) is *valid*, noted $\vDash \boldsymbol{b} \rightarrowtail A$ (resp. $\vDash \boldsymbol{b} \leftarrowtail A$), when $[\![\boldsymbol{b}]\!] \subseteq [\![A]\!]$ (resp. $[\![A]\!] \subseteq [\![\boldsymbol{b}]\!]$). A sequent $\vdash L$ is *valid* when $\vDash L$. As anticipated, the proof system just introduced is sound and complete with respect to the semantics of CPL_0: a labelled formula is valid if and only if it is provable. Soundness can be established by a standard induction on the derivation height. The proof of completeness is less straightforward and described in full detail in [1, § A.2]. The fundamental ingredient is the introduction of a *decomposition relation* between finite sets of sequents, which allows one to decompose the validity of a complex statement (for example, $\boldsymbol{b} \rightarrowtail A \vee B$) into that of a finite set of less complex statements (such as, $\boldsymbol{c} \rightarrowtail A, \boldsymbol{d} \rightarrowtail B$, given that $\boldsymbol{b} \vDash \boldsymbol{c} \vee \boldsymbol{d}$ holds). One can show then that a complex valid sequent is decomposable into a finite set of non-decomposable valid sequents, and, from the provability of the latter, climb back to the validity of the original sequent using the rules of CPL_0.

Proposition 1. $\vDash L$ *holds if and only if* $\vdash_{\mathsf{CPL}_0} L$ *holds*

[1] Observe that the last four counting rules in Fig. 1 make an *arbitrarily chosen* label $\boldsymbol{b}$ appear in the conclusion. Intuitively, this is coherent with the semantics of counting formulas, which are interpreted as either 2^ω or $\emptyset$, which are (resp.) superset or subset of *any* given set.

Initial Sequents

$$\dfrac{\mathscr{b} \vDash x_n}{\vdash \mathscr{b} \rightarrowtail \mathbf{n}}\ \mathsf{Ax1} \qquad\qquad \dfrac{x_n \vDash \mathscr{b}}{\vdash \mathscr{b} \leftarrowtail \mathbf{n}}\ \mathsf{Ax2}$$

Set Rules

$$\dfrac{\vdash \mathscr{c} \rightarrowtail A \qquad \vdash \mathscr{d} \rightarrowtail A \qquad \mathscr{b} \vDash \mathscr{c} \vee \mathscr{d}}{\vdash \mathscr{b} \rightarrowtail A}\ \mathsf{R}^{\rightarrowtail}_{\cup} \qquad \dfrac{\vdash \mathscr{c} \leftarrowtail A \qquad \vdash \mathscr{d} \leftarrowtail A \qquad \mathscr{c} \wedge \mathscr{d} \vDash \mathscr{b}}{\vdash \mathscr{b} \leftarrowtail A}\ \mathsf{R}^{\leftarrowtail}_{\cap}$$

Logical Rules

$$\dfrac{\vdash \mathscr{c} \leftarrowtail A \qquad \mathscr{b} \vDash \neg \mathscr{c}}{\vdash \mathscr{b} \rightarrowtail \neg A}\ \mathsf{R}^{\rightarrowtail}_{\neg} \qquad \dfrac{\vdash \mathscr{c} \rightarrowtail A \qquad \neg \mathscr{c} \vDash \mathscr{b}}{\vdash \mathscr{b} \leftarrowtail \neg A}\ \mathsf{R}^{\leftarrowtail}_{\neg}$$

$$\dfrac{\vdash \mathscr{b} \rightarrowtail A}{\vdash \mathscr{b} \rightarrowtail A \vee B}\ \mathsf{R1}^{\rightarrowtail}_{\vee} \qquad \dfrac{\vdash \mathscr{b} \rightarrowtail B}{\vdash \mathscr{b} \rightarrowtail A \vee B}\ \mathsf{R2}^{\rightarrowtail}_{\vee} \qquad \dfrac{\vdash \mathscr{b} \leftarrowtail A \qquad \vdash \mathscr{b} \leftarrowtail B}{\vdash \mathscr{b} \leftarrowtail A \vee B}\ \mathsf{R}^{\leftarrowtail}_{\vee}$$

$$\dfrac{\vdash \mathscr{b} \leftarrowtail B}{\vdash \mathscr{b} \leftarrowtail A \wedge B}\ \mathsf{R1}^{\leftarrowtail}_{\wedge} \qquad \dfrac{\vdash \mathscr{b} \leftarrowtail A}{\vdash \mathscr{b} \leftarrowtail A \wedge B}\ \mathsf{R2}^{\leftarrowtail}_{\wedge} \qquad \dfrac{\vdash \mathscr{b} \rightarrowtail A \qquad \vdash \mathscr{b} \rightarrowtail B}{\vdash \mathscr{b} \rightarrowtail A \wedge B}\ \mathsf{R}^{\rightarrowtail}_{\wedge}$$

Counting Rules

$$\dfrac{\mu(\llbracket \mathscr{b} \rrbracket) = 0}{\vdash \mathscr{b} \rightarrowtail A}\ \mathsf{R}^{\rightarrowtail}_{\mu} \qquad \dfrac{\mu(\llbracket \mathscr{b} \rrbracket) = 1}{\vdash \mathscr{b} \leftarrowtail A}\ \mathsf{R}^{\leftarrowtail}_{\mu}$$

$$\dfrac{\vdash \mathscr{c} \rightarrowtail A \qquad \mu(\llbracket \mathscr{c} \rrbracket) \geq q}{\vdash \mathscr{b} \rightarrowtail \mathbf{C}^q A}\ \mathsf{R}^{\rightarrowtail}_{\mathbf{C}} \qquad \dfrac{\vdash \mathscr{c} \leftarrowtail A \qquad \mu(\llbracket \mathscr{c} \rrbracket) < q}{\vdash \mathscr{b} \leftarrowtail \mathbf{C}^q A}\ \mathsf{R}^{\leftarrowtail}_{\mathbf{C}}$$

$$\dfrac{\vdash \mathscr{c} \leftarrowtail A \qquad \mu(\llbracket \mathscr{c} \rrbracket) < q}{\vdash \mathscr{b} \rightarrowtail \mathbf{D}^q A}\ \mathsf{R}^{\rightarrowtail}_{\mathbf{D}} \qquad \dfrac{\vdash \mathscr{c} \rightarrowtail A \qquad \mu(\llbracket \mathscr{c} \rrbracket) \geq q}{\vdash \mathscr{b} \leftarrowtail \mathbf{D}^q A}\ \mathsf{R}^{\leftarrowtail}_{\mathbf{D}}$$

Fig. 1. Proof System for CPL_0

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x_0 \vDash x_0}{\vdash x_0 \rightarrowtail \mathbf{0}}\ \mathsf{Ax1}}{\vdash x_0 \wedge \neg x_1 \rightarrowtail \mathbf{0}}\ \mathsf{R}^{\rightarrowtail}_{\cup} \quad \dfrac{\dfrac{\dfrac{x_1 \vDash x_1}{\vdash x_1 \leftarrowtail \mathbf{1}}\ \mathsf{Ax2}}{\vdash \neg x_1 \rightarrowtail \neg \mathbf{1}}\ \mathsf{R}^{\rightarrowtail}_{\neg}}{\vdash x_0 \wedge \neg x_1 \rightarrowtail \neg \mathbf{1}}\ \mathsf{R}^{\rightarrowtail}_{\cup}}{\vdash x_0 \wedge \neg x_1 \rightarrowtail \mathbf{0} \wedge \neg \mathbf{1}}\ \mathsf{R}^{\rightarrowtail}_{\wedge}}{\vdash x_0 \wedge \neg x_1 \rightarrowtail (\mathbf{0} \wedge \neg \mathbf{1}) \vee (\neg \mathbf{0} \wedge \mathbf{1})}\ \mathsf{R1}^{\rightarrowtail}_{\vee} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x_0 \vDash x_0}{\vdash x_0 \leftarrowtail \mathbf{0}}\ \mathsf{Ax2}}{\vdash \neg x_0 \rightarrowtail \neg \mathbf{0}}\ \mathsf{R}^{\rightarrowtail}_{\neg}}{\vdash \neg x_0 \wedge x_1 \rightarrowtail \neg \mathbf{0}}\ \mathsf{R}^{\rightarrowtail}_{\cup} \quad \dfrac{\dfrac{x_1 \vDash x_1}{\vdash x_1 \rightarrowtail \mathbf{1}}\ \mathsf{Ax1}}{\vdash \neg x_0 \wedge x_1 \rightarrowtail \mathbf{1}}\ \mathsf{R}^{\rightarrowtail}_{\cup}}{\vdash \neg x_0 \wedge x_1 \rightarrowtail \neg \mathbf{0} \wedge \mathbf{1}}\ \mathsf{R}^{\rightarrowtail}_{\wedge}}{\vdash \neg x_0 \wedge x_1 \rightarrowtail (\mathbf{0} \wedge \neg \mathbf{1}) \vee (\neg \mathbf{0} \wedge \mathbf{1})}\ \mathsf{R2}^{\rightarrowtail}_{\vee}}{\dfrac{\vdash (x_0 \wedge \neg x_1) \vee (\neg x_0 \wedge x_1) \rightarrowtail (\mathbf{0} \wedge \neg \mathbf{1}) \vee (\neg \mathbf{0} \wedge \mathbf{1})}{\vdash \top \rightarrowtail \mathbf{C}^{\frac{1}{2}}((\mathbf{0} \wedge \neg \mathbf{1}) \vee (\neg \mathbf{0} \wedge \mathbf{1}))}\ \mathsf{R}^{\rightarrowtail}_{\mathbf{C}}\,*}\ \mathsf{R}^{\rightarrowtail}_{\cup}$$

*as $\mu(\llbracket (x_0 \wedge \neg x_1) \vee (\neg x_0 \wedge x_1) \rrbracket) \geq \frac{1}{2}$

Fig. 2. Derivation of $\vdash \top \rightarrowtail \mathbf{C}^{\frac{1}{2}}\big((\mathbf{0} \wedge \neg \mathbf{1}) \vee (\neg \mathbf{0} \wedge \mathbf{1})\big)$ in CPL_0

2.3 CPL_0-Validity is in $\mathsf{P}^{\sharp\mathtt{SAT}}$

As suggested before, a proof that a quantified formula like $\mathbf{C}^q A$ or $\mathbf{D}^q A$ is valid can be seen as obtained by invoking an *oracle*, which provides a suitable measurement $\mu(\llbracket \mathscr{b} \rrbracket)$, for a Boolean formula $\mathscr{b}$. As shown by Lemma 1, these measurements correspond to actually *counting* the number of valuations satisfying the corresponding formula. It is possible to make this intuition precise by showing that, in $\mathbf{CPL}_0$, validity can be decided by a polytime algorithm having access to an oracle for the problem $\sharp$`SAT` of counting the models of a Boolean formula.

A formula of $\mathbf{CPL}_0$, call it A, is said to be *closed* if it is either of the form $\mathbf{C}^q B$ or $\mathbf{D}^q B$ or it is a negation, conjunction, or disjunction of closed formulas. It can be easily checked by induction on the structure of closed formulas that for any closed A, either $[\![A]\!] = 2^\omega$ or $[\![A]\!] = \emptyset$. We define, by mutual recursion, two polytime algorithms `Bool` and `Val`: for each formula A of $\mathbf{CPL}_0$, `Bool`(A) computes a Boolean formula $ƀ_A$ such that $[\![A]\!] = [\![ƀ_A]\!]$, and, for all closed formula A, `Val`$(A) = 1$ if and only if $[\![A]\!] = 2^\omega$ and `Val`$(A) = 0$ if and only if $[\![A]\!] = \emptyset$. The two algorithms are defined in Figure 3. Notice that the algorithm `Val` makes use of a $\sharp$`SAT` oracle.

We recall that the class $\mathsf{P}^{\sharp\mathtt{SAT}}$ is made of those problems which can be decided in polytime having access to a $\sharp$`SAT` oracle. One can easily be convinced that the algorithms `Bool` and `Val` both belong to $\mathsf{P}^{\sharp\mathtt{SAT}}$, which leads to the following:

Proposition 2. $\mathbf{CPL}_0$*-validity is in* $\mathsf{P}^{\sharp\mathtt{SAT}}$.

$$
\begin{aligned}
\mathtt{Bool}(n) &= x_n \\
\mathtt{Bool}(A_1 \wedge A_2) &= \mathtt{Bool}(A_1) \wedge \mathtt{Bool}(A_2) \\
\mathtt{Bool}(A_1 \vee A_2) &= \mathtt{Bool}(A_1) \vee \mathtt{Bool}(A_2) \\
\mathtt{Bool}(\neg A_1) &= \neg \mathtt{Bool}(A_1) \\
\mathtt{Bool}(\mathbf{C}^q A_1) &= \mathtt{Val}(\mathbf{C}^q A_1) \\
\mathtt{Bool}(\mathbf{D}^q A_1) &= \mathtt{Val}(\mathbf{D}^q A_1)
\end{aligned}
$$

$$
\begin{aligned}
\mathtt{Val}(A_1 \wedge A_2) &= \mathtt{Val}(A_1)\ \mathit{AND}\ \mathtt{Val}(A_2) \\
\mathtt{Val}(A_1 \vee A_2) &= \mathtt{Val}(A_1)\ \mathit{OR}\ \mathtt{Val}(A_2) \\
\mathtt{Val}(\neg A_1) &= \mathit{NOT}\ \mathtt{Val}(A_1) \\
\mathtt{Val}(\mathbf{C}^q A_1) &= \ \mathit{let}\ ƀ = \mathtt{Bool}(A_1)\ \mathit{in} \\
&\quad\ \mathit{let}\ n = \sharp\mathtt{Val}(ƀ)\ \mathit{in} \\
&\quad\ \frac{\sharp\mathtt{SAT}(ƀ)}{2^n} \geq q \\
\mathtt{Val}(\mathbf{D}^q A_1) &= \ \mathit{let}\ ƀ = \mathtt{Bool}(A_1)\ \mathit{in} \\
&\quad\ \mathit{let}\ n = \sharp\mathtt{Val}(ƀ)\ \mathit{in} \\
&\quad\ \frac{\sharp\mathtt{SAT}(ƀ)}{2^n} < q
\end{aligned}
$$

where $\sharp\mathtt{Val}(ƀ)$ is the number of propositional variables in $ƀ$.

Fig. 3. `Bool`$(\cdot)$ and `Val`$(\cdot)$

3 On Multivariate Counting Propositional Logic

In this section we introduce propositional counting logic $\mathbf{CPL}$, which extends counting quantifiers to a *multivariate* case, as discussed below. In Section 4 it will be shown that this logic yield a characterization of the full counting hierarchy.

As it is well-known, counting problems are not restricted to those in $\mathsf{P}^{\sharp\mathtt{SAT}}$. For instance, one can consider problems concerning relations between valuations of *different* groups of variables, like MajMajSAT [7,25,26]. Given a formula A of **PL** containing two disjoint sets $\mathbf{x}$ and $\mathbf{y}$ of variables, this problem asks whether for at least half of the valuations of $\mathbf{x}$, at least half of the valuations of $\mathbf{y}$ makes A true.

To express these kinds of problems, we will consider a language in which propositional atoms and counting quantifiers are *named* (we use $a, b, c, \dots$ for names); counting quantifications, indicated as $\mathbf{C}^q_a A$ or $\mathbf{D}^q_a A$, now depend on the number of valuations of propositional atoms *with name* a satisfying A.

Definition 6 (Formulas of CPL). *The* formulas of **CPL** *are defined by the following grammar:*

$$A ::= \mathbf{i}_a \;\big|\; \neg A \;\big|\; A \wedge B \;\big|\; A \vee B \;\big|\; \mathbf{C}^q_a A \;\big|\; \mathbf{D}^q_a A$$

where $i \in \mathbb{N}$, *a is a name, and* $q \in \mathbb{Q}_{[0,1]}$.

Named quantifiers, $\mathbf{C}^q_a$ and $\mathbf{D}^q_a$, bind the occurrences of the name a in A. Given a formula A of **CPL**, we let $\mathrm{FN}(A)$ indicate the set of names occurring *free* (i.e. not bound) in A.

Names can be used to distinguish between distinct groups of propositional variables. For example, the propositional formula $F = (x_1 \vee y_1) \wedge (x_2 \vee y_2)$, containing two groups of variables $\mathbf{x} = \{x_1, x_2\}$ and $\mathbf{y} = \{y_1, y_2\}$, can be expressed in **CPL** using two distinct names a, b as $G = (\mathbf{1}_a \vee \mathbf{1}_b) \wedge (\mathbf{2}_a \vee \mathbf{2}_b)$. Since the intuitive meaning of $\mathbf{C}^q_a A$ is that A is true in at least q of the valuations of the variables with name a, we can take the **CPL**-formula $\mathbf{C}^{\frac{1}{2}}_a \mathbf{C}^{\frac{1}{2}}_b G$ as expressing the MajMajSAT problem for F (which happens to have a positive answer, in this case).

While the formulas $\mathbf{C}^q_a A$ and $\mathbf{D}^q_a A$ have a rather intuitive meaning, the semantics of **CPL**-formulas is slightly subtler than in the case of $\mathbf{CPL}_0$. The interpretation of a formula A now depends on the choice of a finite set of names $X \supseteq \mathrm{FN}(A)$, and consists in a measurable set $[\![A]\!]_X$ belonging to the Borel algebra $\mathcal{B}\big((2^\omega)^X\big)$. Hence, the quantifiers $\mathbf{C}^q_a$ and $\mathbf{D}^q_a$ must correspond to operations allowing one to pass from $\mathcal{B}\big((2^\omega)^{X \cup \{a\}}\big)$ to $\mathcal{B}\big((2^\omega)^X\big)$. To define such operations we need the following technical notion: given two disjoint finite sets of names X, Y, for any $f \in (2^\omega)^X$, and $\mathcal{X} \subseteq (2^\omega)^{X \cup Y}$, the *$f$-projection* of $\mathcal{X}$ is the set $\Pi_f(\mathcal{X}) = \{g \in (2^\omega)^Y \mid f + g \in \mathcal{X}\} \subseteq (2^\omega)^Y$, where $(f+g)(\alpha)$ is $f(\alpha)$, if $\alpha \in X$ and $g(\alpha)$ if $\alpha \in Y$.

Definition 7 (Semantics of CPL). *For each formula A of* **CPL**, *and finite set of names such that* $X \supseteq \mathrm{FN}(A)$, *the* interpretation *of A,* $[\![A]\!]_X \subseteq (2^\omega)^X$, *is inductively defined as follows:*

$$\begin{aligned} [\![\mathbf{i}_a]\!]_X &= \{f \mid f(a)(i) = 1\} & [\![\neg A]\!]_X &= (2^\omega)^X \;-\; [\![A]\!]_X \\ [\![A \wedge B]\!]_X &= [\![A]\!]_X \cap [\![B]\!]_X & [\![\mathbf{C}^q_a A]\!]_X &= \{f \mid \mu\big(\Pi_f([\![A]\!]_{X \cup \{a\}})\big) \geq q\} \\ [\![A \vee B]\!]_X &= [\![A]\!]_X \cup [\![B]\!]_X & [\![\mathbf{D}^q_a A]\!]_X &= \{f \mid \mu\big((\Pi_f([\![A]\!]_{X \cup \{a\}})\big) < q\}. \end{aligned}$$

That all sets $\llbracket A \rrbracket_X$ are measurable, namely that $\llbracket A \rrbracket_X \in \mathcal{B}\big((2^\omega)^X\big)$, is not an obvious fact (as it crucially relies on some properties of f-projections), and is proved in detail in [1, § 4]. Logical equivalence in **CPL** is defined relatively to a set of names X, by letting $A \equiv_X B$ if and only if $\mathrm{FN}(A), \mathrm{FN}(B) \subseteq X$ and $\llbracket A \rrbracket_X = \llbracket B \rrbracket_X$.

Similarly to what has been shown in the previous section, one can introduce a sound and complete labelled calculus for **CPL**. In this case, labelled formulas involve a named Boolean formula (i.e, built from named Boolean variables, such as $\boldsymbol{x}_i^a$). Sequents are of the form $\vdash^X L$, with $\mathrm{FN}(L) \subseteq X$. Most rules of **CPL** are straightforward generalizations of those for $\mathbf{CPL}_0$, except for the counting rules, which are understandably more complex (see the example in Figure 4), and rely on the notion of *a-decomposition* for Boolean formulas.[2] In spite of this involved definition, soundness and completeness for this calculus can be still proved generalizing the corresponding arguments for $\mathbf{CPL}_0$.

$$\frac{\vdash^{X \cup \{a\}} \mathscr{c} \rightarrowtail A \qquad \mathscr{b} \vDash^X \bigvee_i \{ \mathscr{e}_i \mid \mu(\llbracket \mathscr{d}_i \rrbracket_{\{a\}}) \geq q \}}{\vdash^X \mathscr{b} \rightarrowtail \mathbf{C}_a^q A} \mathsf{R}_{\overrightarrow{\mathbf{C}}} *$$

* where $\bigvee_i \mathscr{e}_i \wedge \mathscr{d}_i$ is an a-decomposition of $\mathscr{c}$

Fig. 4. Example of Counting Rule Schema for **CPL**

4 Relating CPL to the Counting Hierarchy

We have already seen that the problem `MajMajSAT`, which is complete for $\mathsf{CH}_2 = \mathsf{PP}^{\mathsf{PP}}$, is "captured" by formulas of the form $\mathbf{C}_a^q \mathbf{C}_b^r A$, where A is quantifier-free. We will extend this result to all levels of CH by considering **CPL**-formulas containing an arbitrary number of counting quantifiers. We will proceed in three steps. First, we will show that any formula of **CPL** can be put in *prenex normal form*, that is, that all counting quantifiers can be moved at top-level. Next, we will prove that the **D** quantifier, which has no counterpart in Wagner's problems, can be eliminated. Finally, using Wagner's Theorem [40], we will show that prenex formulas with k nested **C**-quantifiers characterize the level k of CH.

[2] Given a named Boolean formula $\mathscr{b}$, with free names in $X \cup \{a\}$, an *a-decomposition* of $\mathscr{b}$ is any Boolean formula $\mathscr{c} = \bigvee_{i=0}^{k-1} \mathscr{d}_i \wedge \mathscr{e}_i$ such that: (i) $\llbracket \mathscr{c} \rrbracket_{X \cup \{a\}} = \llbracket \mathscr{b} \rrbracket_{X \cup \{a\}}$, (ii) $\mathrm{FN}(\mathscr{d}_i) \subseteq \{a\}$ and $\mathrm{FN}(\mathscr{e}_i) \subseteq X$, (iii) if $i \neq j$, then $\llbracket \mathscr{e}_i \rrbracket_X \cap \llbracket \mathscr{e}_j \rrbracket_X = \emptyset$. It can be shown that any Boolean formula $\mathscr{b}$, with $\mathrm{FN}(\mathscr{b}) \subseteq X \cup \{a\}$ admits an a-decomposition. For further details, see [1, § 4]. The complete proof system for **CPL** is presented in [1, § B.2].

4.1 On Wagner's Counting Hierarchy

The existence of deep and mutual interactions between classical propositional logic and computational complexity is well-known. For instance, checking the satisfiability of **PL**-formulas is the paradigmatic NP-complete problem [11], while the subclass of all tautologies is coNP-complete. When switching to **QPL**, these classes are even captured by a single logical concept: already in the early 1970s, Meyer and Stockmeyer defined PH, and proved that each level of the hierarchy is characterized by the validity of prenex QBF with the corresponding number of quantifier alternations [27, 28, 35, 42, 10].

Nevertheless, if we move to a probabilistic framework, such a plain correspondence seems lost, as no analogous *logical* counterpart has yet been found for the counting complexity classes and hierarchy, introduced from the 1970s on by Valiant [38] and Wagner [39–41]. Specifically, CH was presented in 1986 (actually by both Wagner and Parberry and Schnitger [33]) as the counting analogous to PH, which is inadequate to characterize problems in which counting is involved. The hierarchy CH is defined similarly to PH, but with PP in place of NP, i.e. by letting $\mathsf{CH}_0 = \mathsf{P}$ and $\mathsf{CH}_{n+1} = \mathsf{PP}^{\mathsf{CH}_n}$. Actually, the original characterization in [40] is in terms of alternating (counting and standard) quantifiers, where Wagner's counting operator is capable of expressing also standard quantification. In [40], beyond introducing CH, the author also considers canonical complete problems for each level of the hierarchy.

In the rest of this section we will relate such results to **CPL**, showing that the validity of the formulas of **CPL** yields a new family of complete problems for CH_n, hence providing a logical characterization of CH.

4.2 Prenex Normal Forms

Let us introduce prenex normal forms in the language of **CPL**:

Definition 8 (PNF). *A formula of* **CPL** *is an n-ary* prenex normal form *(or simply a* prenex normal form*, PNF for short) if it can be written as $\triangle_1 \dots \triangle_n A$, where, for every $i \in \{1, \dots, n\}$, $\triangle_i$ is either $\mathbf{C}^q_a$ or $\mathbf{D}^q_a$ (for arbitrary a and q), and A is quantifier-free. The formula A is said to be the* matrix *of the PNF.*

To convert a formula of **CPL** into an equivalent PNF, some intermediate lemmas are needed.[3] Preliminarily, notice that for every formula A of **CPL**, name a, and finite set X, such that $\mathrm{FN}(A) \subseteq X$ and $a \notin X$, if $q = 0$, then $[\![\mathbf{C}^q_a A]\!]_X = (2^\omega)^X$ and $[\![\mathbf{D}^q_a A]\!]_X = \emptyset^X$.

The lemma below shows that counting quantifiers occurring inside any conjunction or disjunction can be extruded from it.

Lemma 2. *Let $a \notin \mathrm{FN}(A)$ and $q > 0$. Then, for every X such that $\mathrm{FN}(A) \cup \mathrm{FN}(B) \subseteq X$ and $a \notin X$, the following equivalences hold:*

$$A \wedge \mathbf{C}^q_a B \equiv_X \mathbf{C}^q_a (A \wedge B) \qquad A \vee \mathbf{C}^q_a B \equiv_X \mathbf{C}^q_a (A \vee B)$$
$$A \wedge \mathbf{D}^q_a B \equiv_X \mathbf{D}^q_a (\neg A \vee B) \qquad A \vee \mathbf{D}^q_a B \equiv_X \mathbf{D}^q_a (\neg A \wedge B).$$

[3] Their proofs can be found in [1, § C].

Remarkably, a corresponding lemma does *not* hold for CPL_0, due to the impossibility of *renaming* variables (on which Lemma 2 relies).

We then consider negation. In this case, the inter-definability of $\mathbf{C}^q$ and $\mathbf{D}^q$ in CPL_0 (Equation 1) can be generalized to CPL, and this allows one to get rid of negations which lie between any occurrences of a counting quantifier and the formula's root.

Lemma 3. *For every $q \in \mathbb{Q}_{[0,1]}$, name a, and X such that $\mathrm{FN}(A) \subseteq X \cup \{a\}$, and $a \notin X$, $\neg\mathbf{D}^q_a A \equiv_X \mathbf{C}^q_a A$ and $\neg\mathbf{C}^q_a A \equiv_X \mathbf{D}^q_a A$ hold.*

Therefore, using Lemma 2 and Lemma 3, we can conclude that every formula of CPL can be put in PNF, as desired.

Proposition 3. *For every formula A of CPL there is a PNF B, such that for every X with $\mathrm{FN}(A)\cup\mathrm{FN}(B) \subseteq X$, $A \equiv_X B$ holds. Moreover, B can be computed in polynomial time from A.*

4.3 Positive Prenex Normal Forms

Reducing formulas to PNF is close to what we need, but there is one last step to make, namely getting rid of the quantifier $\mathbf{D}$, which does not have any counterpart in Wagner's construction. In other words, we need to reduce CPL-formulas to prenex normal forms *of a special kind*:

Definition 9 (PPNF). *A formula of CPL is said to be a* positive prenex normal form *(PPNF, for short) when it is both PNF and $\mathbf{D}$-free.*

The gist to convert formulas into (equivalent) PPNF, consists in two main steps: (i) converting each instance of $\mathbf{D}$ into one of $\mathbf{C}$, using Lemma 3, and (ii) applying the lemma below which states that $\mathbf{C}$ enjoys a specific, weak form of self duality, to push the negation inside the matrix.

Lemma 4 (Epsilon Lemma). *For every formula A of CPL and $q \in \mathbb{Q}_{[0,1]}$, there is a $p \in \mathbb{Q}_{[0,1]}$ such that, for every X with $\mathrm{FN}(A) \subseteq X$ and $a \notin X$: $\neg\mathbf{C}^q_a A \equiv_X \mathbf{C}^p_a \neg A$. Moreover, p can be computed from q in polynomial time.*

Proof (Sketch[4]). Let $\mathscr{b}_A$ be a Boolean formula satisfying $[\![A]\!]_{X\cup\{a\}} = [\![\mathscr{b}_A]\!]_{X\cup\{a\}}$, a-decomposable as $\bigvee_i^n \mathscr{d}_i \wedge \mathscr{e}_i$, and let k be maximum such that $\boldsymbol{x}^a_k$ occurs in $\mathscr{b}_A$. Let $[0,1]_k$ be the set of those rational numbers $r \in [0,1]$ which can be written as a finite sum of the form $\sum_{i=0}^k b_i \cdot 2^i$. For all $i \in \{0,\dots,n\}$, $\mu([\![\mathscr{d}_i]\!]_{\{a\}}) \in [0,1]_k$, where $b_i \in \{0,1\}$, and for all $f : X \to 2^\omega$, also $\mu\big(\Pi_f([\![A]\!]_{X\cup\{a\}})\big) \in [0,1]_k$. Let now ϵ be $2^{-(k+1)}$ if $q \in [0,1]_k$ and $q \neq 1$, ϵ be $2^{-(k+1)}$ if $q = 1$ and let $\epsilon = 0$ if $q \notin [0,1]_k$. In all cases, $q+\epsilon \notin [0,1]_k$ so, by means of some simple computation, it is possible to conclude that $[\![\neg\mathbf{C}^q_a A]\!]_X = [\![\mathbf{C}^{1-(q+\epsilon)}_a \neg A]\!]_X$.

Actually, the value of p is very close to $1-q$, the difference between the two being easily computable from the formula A. So, any negation occurring in the counting prefix of a PNF formula, can be pushed back into the matrix.

[4] For full details, see [1, Lemma 13].

Proposition 4. *For every formula A of* **CPL** *there is a PPNF B such that for every X, with* $\mathrm{FN}(A) \cup \mathrm{FN}(B) \subseteq X$, $A \equiv_X B$ *holds. Moreover, B can be computed from A in polynomial time.*

4.4 CPL and the Counting Hierarchy

As anticipated, in [40] Wagner not only introduced his counting operator and hierarchy, but also defined complete problems for *each level* of CH. Below, we present a slightly weaker version of Wagner's Theorem [40, pp. 338-339], which perfectly fits our needs.

Suppose $\mathcal{L}$ to be a subset of S^n, where S is a set, that $1 \leq m < n$, and that $b \in \mathbb{N}$. We define $\mathbf{C}^b_m\mathcal{L}$ as the following subset of S^{n-m}:

$$\{(a_n, \dots, a_{m+1}) \mid \#(\{(a_m, \dots, a_1) \mid (a_n, \dots, a_1) \in \mathcal{L}\}) \geq b\}\,.$$

Let $\mathbf{T}$ and $\mathbf{F}$ indicate the usual true and false formulas of **PL**. For any natural number $n \in \mathbb{N}$, let $\mathcal{TF}^n$ be the subset of PL^{n+1} containing all tuples in the form $(A, t_1, \dots, t_n)$, where A is a propositional formula in CNF with at most n free variables, and $t_1, \dots, t_n \in \{\mathbf{T}, \mathbf{F}\}$ render A true. Finally, for every $k \in \mathbb{N}$, we denote as W^k the language consisting of all (binary encodings of) tuples of the form $(A, m_1, \dots, m_k, b_1, \dots, b_k)$ such that $A \in \mathbf{C}^{b_1}_{m_1} \cdots \mathbf{C}^{b_k}_{m_k} \mathcal{TF}^{\sum m_i}$.

Theorem 1 (Wagner, Th.7 [40]). *For every k, the language W^k is complete for CH_k.*

Observe that elements of W^k can be seen as alternative representations for PPNF formulas of **CPL**, once any m_i is replaced by $\min\{1, \frac{m_i}{2^{b_i}}\}$. Consequently,

Corollary 1. *The closed and valid k-ary PPNFs, whose matrix is in CNF, define a complete set for CH_k.*

5 Related Works

The literature on logics enabling some forms of probabilistic reasoning is vast, yet most proposals are not related to computational aspects. In the last decades, several probabilistic logics have been developed in the realm of modal logic, starting from the pioneering works by Nilsson [30, 31]. In particular, in the 1990s, some noteworthy probability logics were (independently) introduced both by Bacchus [6, 4, 5] and by Fagin, Halpern, and Megiddo [14, 18, 13, 19]. Another class of probabilistic modal logics have been designed to model Markov chains and similar structures, see for instance [20, 23, 24]. A notable example is *Riesz modal logic* [15], which admits a sound and complete proof system. Remarkably, this is the only sequent calculus for probability logic we are aware of, while complete axiomatic systems have been provided for both the probability logics quoted above [6, 14]. By the way, our calculi are actually inspired by labelled systems, such as $\mathbf{G3K}^*$ and $\mathbf{G3P}^*$, as presented for example in [29, 17].

As we have seen, CH was first defined by Wagner in [39, 41, 40] and, independently, by Pareberry and Schnitger [33]. It was conceived as an extension of Meyer and Stockmeyer's PH [27, 28] aiming at characterizing natural problems in which counting is involved. There are two main, equivalent [37] ways to define CH: the original characterization in terms of *alternating quantifiers* [40], and the one based on *oracles* [36]. Notably, Wagner's operator was not the only "probabilistic" (class) quantifiers introduced in the 1980s (consider, for instance, Papadimitriou's *probabilistic* quantifier [32], Zachos and Heller's *random* quantifier [44], or Zachos' *overwhelming* and *majority* quantifiers [43]). However, to the best of the authors' knowledge, all these operators are counting quantifiers on (classes of) languages, rather than *stricto sensu* logical ones. One remarkable exception is represented by Kontinen's work [22], in which second-order quantifiers are defined in the style of descriptive complexity.

6 Conclusion

To the best of our knowledge, **CPL** is the first logical system extending propositional logic with counting quantifiers. Our main source of inspiration comes from computational complexity, namely from Wagner's counting operator on classes. By the way, we believe that the main contribution of the paper is not the introduction of counting logics *per se*, but the investigation of its connections with counting classes. Indeed, we have shown that counting quantifiers play nicely with propositional logic in characterizing CH, and thus relate nicely with some old and recent results in complexity theory. In our opinion, **CPL** naturally appears as the probabilistic counterpart of **QPL**.

Due to space reasons, we left out some important applications of counting logics to other branches of computer science, such as the theory of programming languages. In particular, it is possible to design type systems for the randomized λ-calculus by extending simple types with counting quantifiers,[5] and to define a probabilistic counterpart of the Curry-Howard correspondence [16, 34] relating typing derivations with derivations in **CPL**.[6] Moreover, the proof theory of **CPL** has just been briefly delineated and the dynamics (i.e. the cut-elimination procedure) of the introduced formal systems deserves further investigation. Promising results also concern the possibility to inject "counting" quantifiers into the language of arithmetic. In particular, in [2] we have investigated an extension of standard Peano Arithmetics with *measure quantifiers*, which can be seen as a natural generalization of the quantifiers of $\mathbf{CPL}_0$ to the language of arithmetic. The extension of counting quantifiers to arithmetic looks particularly promising, as it suggests ways of characterizing in a "logical" way explicit lower bounds for counting problems [26], as well as the possibility of defining new logical systems capturing probabilistic complexity classes like BPP (see [21]).

[5] Notice that while several type systems for randomized λ-calculi and guaranteeing various forms of termination properties have been introduced in the last years, [12, 9, 3], none of these systems is explicitly logic-oriented.

[6] Some achievements in this direction have been presented in [1, § 6].

References

1. Antonelli, M., Dal Lago, U., Pistone, P.: On Counting Propositional Logic (2021), available at: https://arxiv.org/abs/2103.12862
2. Antonelli, M., Dal Lago, U., Pistone, P.: On Measure Quantifiers in First-Order Arithmetic (2021), to appear in *Proceedings of Computability in Europe 2021 (CiE2021)*; long version available at: https://arxiv.org/abs/2104.12124
3. Avanzini, M., Dal Lago, U., Ghyselen, A.: Type-based complexity analysis of probabilistic functional programs. In: Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). pp. 1–13. IEEE, Vancouver, BC, Canada, Canada (2019)
4. Bacchus, F.: Lp, a logic for representing and reasoning with statistical knowledge. Computational Intelligence **6**(4), 209–231 (1990)
5. Bacchus, F.: On probability distributions over possible worlds. Machine Intelligence and Pattern Recognition **9**, 217–226 (1990)
6. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge. MIT Press (1990)
7. Biere, A., Heule, M., van Maaren, H., Walsh, T.: Handbook of Satisfiability. IOS Press (2009)
8. Billingsley, P.: Probability and Measure. Wiley (1995)
9. Breuvart, F., Dal Lago, U.: On intersection types and probabilisitic lambda calculi. In: PPDP '18: Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming. pp. 1–13. No. 8 (2018)
10. Büning, H., Bubeck, U.: Theory of quantified Boolean formulas. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability. IOS Press (2009)
11. Cook, S.: The complexity of theorem-proving procedures. In: STOC '71. pp. 151–158 (1971)
12. Dal Lago, U., Grellois, U.: Probabilistic termination by monadic affine sized typing. ACM Trans. Program. Lang. Syst. **41**(2), 10–65 (2019)
13. Fagin, R., Halpern, J.: Reasoning about knowledge and probability. Journal of ACM **41**(2), 340–367 (1994)
14. Fagin, R., Halpern, J., Megiddo, N.: A logic for reasoning about probabilities. Inf. Comput. **87**(1/2), 78–128 (1990)
15. Furber, R., Mardare, R., Mio, M.: Probabilistic logics based on Riesz spaces. LMCS **16**(1) (2020)
16. Girard, J.Y.: Proof and Types. Cambridge University Press (1989)
17. Girlando, M., Negri, S., Sbardolini, G.: Uniform labelled calculi for conditional and counterfactual logics. In: WoLLIC 2019. pp. 248–263 (2019)
18. Halpern, J.: An analysis of first-order logics for probability. Artificial Intelligence **46**(3), 311–350 (1990)
19. Halpern, J.: Reasoning About Uncertainty. MIT Press (2003)
20. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Form. Asp. Comput. **6**(5), 512–535 (1994)
21. Jerábek, E.: Approximate counting in Bounded Arithmetic. J. Symb. Log. **72**(3), 959–993 (2007)
22. Kontinen, J.: A logical characterization of the Counting Hierarchy. TOCL (2009)
23. Kozen, D.: Semantics of probabilistic programs. JCSS **53**(3), 165–198 (1982)
24. Lehmann, D., Shelah, S.: Reasoning with time and chance. Inf. Control. **53**(3), 165 – 198 (1982)

25. van Melkebeek, D.: A survey on lower bounds for satisfiability and related problems. FnT-TCS **2**, 197–303 (2007)
26. van Melkebeek, D., Watson, T.: A Quantum Time-Space Lower Bound for the Counting Hierarchy, available at: https://minds.wisconsin.edu/handle/1793/60568
27. Meyer, A., Stockmeyer, L.: The equivalence problem for regular expressions with squaring requires exponential space. In: SWAT. pp. 125–129 (1972)
28. Meyer, A., Stockmeyer, L.: Word problems requiring exponential time (preliminary report). In: STOC'73. pp. 1–9 (1973)
29. Negri, S., von Plato, J.: Proof Analysis: A Contribution to Hilbert's Last Problem. Cambridge University Press (2011)
30. Nilsson, N.: Probabilistic logic. Artificial Intelligence **28**(1), 71–87 (1986)
31. Nilsson, N.: Probabilistic logic revisited. Artificial Intelligence **59**(1/2), 39–42 (1993)
32. Papadimitriou, C.: Games against nature. JCSS **31**(2), 288–301 (1985)
33. Parberry, I., Schnitger, G.: Parallel computation with threshold functions. JCSS **36**, 278–302 (1988)
34. Sorensen, M., Urzyczyn, P.: Lectures on the Curry-Howard Isomorphism, vol. 149. Elsevier (2006)
35. Stockmeyer, L.: The Polynomial-Time Hiearchy. Theor. Comput. Sci. **3**, 1–22 (1977)
36. Torán, J.: An oracle characterization of the Counting Hierarchy. In: Proceedings. Structure in Complexity Theory Third Annual Conference. pp. 213–223 (1988)
37. Torán, J.: Complexity classes defined by counting quantifiers. Journal of the ACM **38**(3), 753–774 (1991)
38. Valiant, L.: The complexity of computing the permanent. Theor. Comput. Sci. **8**(2), 189–201 (1979)
39. Wagner, K.: Compact descriptions and the counting polynomial-time hierarchy. In: Frege Conference 1984: Proceedings of the International Conference held at Schwerin. pp. 383–392 (1984)
40. Wagner, K.: The complexity of combinatorial problems with succinct input representation. Acta Informatica **23**, 325–356 (1986)
41. Wagner, K.: Some observations on the connection between counting and recursion. Theor. Comput. Sci. **47**, 131–147 (1986)
42. Wrathall, C.: Complete sets and the Polynomial-Time Hierarchy. Theor. Comput. Sci. **3**(1), 23–33 (1976)
43. Zachos, S.: Probabilistic quantifiers and games. JCSS **36**(3), 433–451 (1988)
44. Zachos, S., Heller, H.: A decisive characterization of BPP. Information and Control pp. 125–135 (1986)