Universidade de Aveiro Departamento de Matemática
2016

Fábio Daniel Moreira Barbosa

Lógica Proposicional Probabilística
Probabilistic Propositional Logic

Universidade de Aveiro Departamento de Matemática
2016

Fábio Daniel Moreira Barbosa

Lógica Proposicional Probabilística
Probabilistic Propositional Logic

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática e Aplicações, realizada sob a orientação científica do Doutor Manuel António Gonçalves Martins, Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Doutor Pedro Filipe Pessoa Macedo** Professor Auxiliar, Universidade de Aveiro
vogais / examiners committee	**Doutor Luís Manuel Dias Coelho Soares Barbosa** Professor Associado, Universidade do Minho
	Doutor Manuel António Gonçalves Martins Professor Auxiliar, Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

Em primeiro lugar agradeço ao meu orientador, Doutor Manuel António Martins, por todo o apoio e entusiasmo demonstrados desde o início, a disponibilidade total e por nunca duvidar das minhas capacidades para a execução deste trabalho.

Ao diretor do Mestrado, Doutor Agostinho Agra, por toda a paciência e disponibilidade demonstradas em esclarecer-me nos momentos mais conturbados do mestrado. Agradeço também à estrutura interna do Departamento de Matemática da Universidade de Aveiro por me ter facultado as instalações adequadas para a escrita desta dissertação.

Pelo incentivo, agradeço aos meus pais e irmão, e à Luisa pelo companheirismo, pela confiança e apoio incondicional.

Agradeço ainda à Diana pela disponibilidade demonstrada na revisão da dissertação. E a todos aqueles que direta ou indiretamente fizeram parte da minha formação académica, o meu muito obrigado!

palavras-chave

Espaços de Probabilidades, Sistemas Lógicos, Cálculo de Hilbert, Lógica Proposicional Probabilística, Correção, Completude, Incerteza, Condicional

resumo

O termo Lógica Probabilística, em geral, designa qualquer lógica que incorpore conceitos probabilísticos num sistema lógico formal. Nesta dissertação, o principal foco de estudo é uma lógica probabilística (designada por *Lógica Proposicional Probabilística Exógena*), que tem por base a Lógica Proposicional Clássica. São trabalhados sobre essa lógica probabilística a síntaxe, a semântica e um cálculo de Hilbert, provando-se diversos resultados clássicos de Teoria de Probabilidade no contexto da EPPL. São também estudadas duas propriedades muito importantes de um sistema lógico - correção e completude. Prova-se a correção da EPPL da forma usual, e a completude fraca recorrendo a um algoritmo de satisfazibilidade de uma fórmula da EPPL. Serão também considerados na EPPL conceitos de outras lógicas probabilísticas (incerteza e probabilidades intervalares) e Teoria de Probabilidades (condicionais e independência).

keywords Probability Spaces, Logic Systems, Hilbert Calculus, Probabilistic Propositional Logic, Soundness, Completeness, Uncertainty, Conditional

abstract The term Probabilistic Logic generally refers to any logic that incorporates probabilistic concepts in a formal logic system. In this dissertation, the main focus of study is a probabilistic logic (called *Exogenous Probabilistic Propositional Logic*), which is based in the *Classical Propositional Logic*. There will be introduced, for this probabilistic logic, its syntax, semantics and a Hilbert calculus, proving some classical results of Probability Theory in the context of EPPL. Moreover, there will also be studied two important properties of a logic system - soundness and completeness. We prove the EPPL soundness in a standard way, and weak completeness using a satisfiability algorithm for a formula of EPPL. It will be considered in EPPL concepts of other probabilistic logics (uncertainty and intervalar probability) and of Probability Theory (independence and conditional).

Contents

List of Tables

Chapter 1

Introduction

Probability Theory and Logic are two of the main tools in the formal study of reasoning, and have been intensively applied in a huge amount of different areas. This work aims to propose a way to combine both of these fields. The topic of probabilistic logic does not appear in standards neither of Logic nor Probability Theory.

There is a wide variety of studies in the literature which have referred *Probabilistic Logic* in different contexts and, because of that, there is not a standard syntax for probabilistic logic. So, in this work, it will be defined a probabilistic logic without relying on any prior knowledge of the reader on this subject.

First, it is important to understand what is a probabilistic logic. As its name induce, probabilistic logics combine the deductive capability of logic systems with the well-founded Probability Theory. One major issue of this combination of two formal theories is that we need to accommodate the continuous nature of probabilities with the discrete setting of a logic system. That issue is probably the most difficult to overcome.

There are several ways to introduce probabilities into a given logic. We can assign probabilities to either formulas or models of a logic. We may modify completely a logic by adding probabilities creating a new one, or instead, we can introduce probabilities in a meta-level, leaving the basic logic unchanged. In this work, our idea is to make the *probabilization* in the most intuitive way possible, by introducing probabilities in the formulas for the *Classical Propositional Logic* (CPL) in a different level (leaving the CPL unchanged).

Initially, we have three central problems to solve. First, how to define the syntax of this logic in order to accommodate probabilities? Second, how to change the semantics of CPL in order to produce probabilistic models from CPL models? And finally, how to develop a proof system that allows reasoning about probabilities and real numbers?

Regarding the syntax of the logic, we will understand, in a way, CPL formulas as probabilistic events which have probabilities associated. Both syntactically and semantically, our approach will be an exogenous approach, that is, we will keep CPL formulas and models unchanged, and will further add some relevant probabilistic logic structure over it.

In computer science, there are many applications where reasoning about probabilities is extremely important. For example, probabilistic programs ([CCFMS07]) and automata ([Rab63], [Sto08]), satisfaction ([AP01]) and model-checking algorithms ([Hen09]). Thus, because of that, introducing probabilistic concepts in formal logic is necessary.

Nowadays, reasoning about probabilistic systems has a huge importance in many different fields (security, artificial intelligence, traffic analysis, among others). Probabilistic logics help formalizing (in a logical context) concepts of Probability Theory, so that there can be applied computational tools of logic to probabilistic problems. For example, in [Nil93] and [FH94] we could see how probabilistic logic can be applied to Artificial Intelligence. And in [HRWW11], we can better analyze the computational applicability of probabilistic logic in probabilistic networks.

Although we might think that probabilistic logics is a recent topic of reasoning, the origin of this idea goes back to the first half of the last century ([Ram26], [Car50]). Kolmogorov was the first to axiomatize probabities ([Kol33]) and his axioms are still cited nowadays, even in this work. Since then, the term *probabilistic logic* has been getting more attention and has grown into several different approaches.

Assigning probabilities to formulas (of some logic), without changing its formal language, has been done in some scientific works (such as in [Nil86], [Nil93]). In addition to assigning probabilities to propositonal formulas, in Adams and Hailperin works ([Ada98], [Hai86], [Hai96]), we have some ideas on how to get conditional formulas and probabilities in it into an elementary probabilistic propositional logic. In addition, there are plenty of other authors who who have studied probabilistic logic by assigning probabilities to formulas (*e.g.* [Fag90], [Haj01]).

In literature, there already exists other probabilistic logics. Namely, probabilistic temporal logics (*e.g.* [BdA95], [ASB$^+$95], [CIN05]), probabilistic dynamic logics (*e.g.* [Koz85], [VPKS13]), and even dynamic probabilistic epistemic logics ([Mor11]), among others. In some works of Mateus and Baltazar ([BMNP07], [BM09]) it is introduced an *exogenous* probabilistic linear temporal logic (EPLTL) that is obtained by enriching a probabilistic propositional logic with linear temporal modalities.

A possible generalization of probabilistic logic is Quantum Logic. In general, it consists in a set of rules for reasoning about propositions that takes the principles of Quantum Theory into account. In [MS04] and [BRS06], an *exogenous* quantum proposicinal logic is studied (EQPL), which is a kind of generalization of EPPL to quantum theory.

Outline

This work is partitioned into several chapters and sections. **Chapter 2** aims to make this text self-contained, *i.e.* it contains the necessary knowledge to understand this work. There are defined all the needed concepts of Probability Theory and Logic. In particular, the usual concepts related to probability spaces (Section 2.1) are defined in this chapter, because our probabilistic logic will make use of them in their models. In a logical context, first we define logic systems (Section 2.2), and Hilbert calculus (Subsection 2.2.1), since we want to make our logic well defined. Soundness and completeness are also defined because one of our goals is to verify if our logic has these properties. Finally, we define *Classical Propositional Logic* (Subsection 2.2.2) and a logic system to real algebraic numbers (Section 2.3) to show that they may be used in the probabilistic logic.

In **Chapter 3**, it is defined the probabilistic logic that it is approached in this thesis, *i.e.* a probabilistic logic which is based on classical propositional logic together with probability spaces: *Exogenous Probabilistic Propositional Logic* (EPPL). It is formulated a syntax for this logic (Section 3.1), which keeps unchanged the base logic (CPL) and introduces probabilities at a different level in its syntax. There is indroduced a semantical interpretation (Section

3.2) and a Hilbert calculus (Section 3.3), and are proved some interesing results that seem like the ones that can be proved in Probability Theory. Finally, are analysed the soundness (Section 3.4) and the completeness (Section 3.5) of this probabilistic logic. We will prove that EPPL is sound in the standard way, and in order to prove that EPPL is weakly complete, we will use a satisfaction algorithm that decides the satisfiability of a given EPPL formula. This algorithm will return an EPPL model that satisfies that formula, if there is such model.

All sections of **Chapter 4** (except the last one) aim to study some concepts that exist in other probabilistic logics and in Probability Theory into the probabilistic propositional logic that we have worked in previous chapter. We will try to prove some interesting properties related to this probability features. Namely, in Section 4.3 we introduce the concept of conditional in EPPL, and there are discussed two ways to do this: the first one consists in introducing the conditional at the probabilities level (keeping unchanged the base logic); and the other consists of putting conditional formulas as formulas in the base logic (now designated by *Supposicional logic*) and the probabilities at a higher level on this new logic. Finally, in order to conclude, in Section 4.4 will be presented a generalization of the probabilization process of propositional logic (done in Chapter 3), but now for any logic system that we want to make probabilistic.

Chapter 2

State of the Art

This chapter aims at making this text self-contained and allowing the reader to understand the remaining chapters of the thesis. With the goal of defining a probabilistic logic, it will first be necessary to introduce some concepts about Probability Theory and Logic.

2.1 Probability Spaces

This section introduces the topological concept of probability space, and some properties of that will be necessary in the following chapters. In order to define probability spaces, we must first define its three constituents: sample space, σ-algebra and probability function.

Definition 2.1.1. The **sample space**, usually denoted as Ω, is the set of outcomes of an random experiment (with $\Omega \neq \emptyset$). An element of the sample space ($\omega \in \Omega$) is called a **sample point** and a subset of the sample space ($E \subseteq \Omega$) is called an **event**.

Definition 2.1.2. Given a sample space Ω, a **σ-algebra** $\mathcal{F}$ over Ω is a collection of subsets of Ω with the following properties:

(i) $\emptyset \in \mathcal{F}$ (*i.e.* emptyset belongs to $\mathcal{F}$);

(ii) if $F \in \mathcal{F}$, then $(\Omega \backslash F) \in \mathcal{F}$ (*i.e.* $\mathcal{F}$ is closed under complementation);

(iii) if $F_i \in \mathcal{F}$ for all $i \in I$, with I a countable set, then $(\cup_{i \in I} F_i) \in \mathcal{F}$ (*i.e.* $\mathcal{F}$ is closed under countable unions).

The most common σ-algebra is the ***powerset*** of Ω (denoted as $\mathcal{P}(\Omega)$), which contains every subset of Ω. There are other interesting σ-algebras to be analyzed, but in the context of this work, we shall focus only on this one.

Definition 2.1.3. Given a σ-algebra over a sample space Ω, a **probability function** is a function $\mathbf{P} : \mathcal{F} \to \mathbb{R}$ that satisfies the following properties:

(i) $\mathbf{P}(F) \geq 0$, for every $F \in \mathcal{F}$;

(ii) $\mathbf{P}(\Omega) = 1$;

(iii) if $(F_i)_{i\in I} \in \mathcal{F}$ (I countable set) is a countable collection of pairwise disjoint sets (that is, $F_i \cap F_j = \emptyset$ for all $i \neq j$), then $\mathbf{P}(\cup_{i\in I} F_i) = \sum_{i\in I} \mathbf{P}(F_i)$.

Definition 2.1.4. A triple $(\Omega, \mathcal{F}, \mathbf{P})$ is called a **probabitity space**, where Ω is a sample space, $\mathcal{F}$ is a σ-algebra over Ω and $\mathbf{P}$ is a probability function.

With the propose of define our probabilistic logic (EPPL), we still need to understand what is a Bernoulli's random variable, and define the concept of stochastic process.

Definition 2.1.5. Let $(\Omega, \mathcal{F}, \mathbf{P})$ be a probability space and S some countable set of real numbers. A function $X : \Omega \to S$ is called a **random variable** (r.v.) if for each $A \subseteq S$, we have $\{\omega \in \Omega : X(\omega) \in A\} \in \mathcal{F}$.

The random variable that will be most used in this work is a Bernoulli random variable, that is a random variable defined in the set $S = \{0, 1\}$. The next definition ends this section on Probability Theory, and completes all the concepts that we will need in this work.

Definition 2.1.6. Given a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ and an arbitrary set T, a **stochastic process** is a function $X : T \times \Omega \to \mathbb{R}$, where for each $t \in T$, $X_t := X(t, \omega)$ is a random variable. A stochastic process is represented by $X = (X_t)_{t\in T}$.

2.2 Logic Systems

Formal logic is a discipline centered in the study of reasoning, inference and, in general, in the different ways new knowledge can be legitimately acquired from previous knowledge.

In this section will be defined the most elementar concepts related with logic systems. The main goal is the definition of a probabilistic logic system in Chapter 3. We will begin by formally define logic systems and satisfaction systems.

Definition 2.2.1. A **logic system** (or **deductive system**) is a pair $\mathscr{L} = (L, \vdash)$, where $L \neq \emptyset$ is a language and $\vdash \subseteq \{0, 1\}^L \times L$ is a consequence operator, that satisfies the following conditions:

(i) if $\delta \in \Delta$, then $\Delta \vdash \delta$ (reflexivity);

(ii) if $\Delta \subseteq \Gamma$ and $\Delta \vdash \delta$, then $\Gamma \vdash \delta$ (weakening);

(iii) if $\Delta \vdash \delta_1$ for all $\delta_1 \in \Gamma$ and $\Gamma \vdash \delta_2$, then $\Delta \vdash \delta_2$ (cut).

If whenever $\Delta \vdash \delta$ there exists a finite set of formulas $\Delta_0 \subseteq \Delta$ such that $\Delta_0 \vdash \delta$, we say that the logic is an **axiomatizable logic system** (*i.e.* a logic system with the finitary property).

Definition 2.2.2. A **satisfaction relation** is a relation $\vDash \subseteq \mathcal{M} \times L$, where $\mathcal{M}$ is the class of models , *i.e.* structures that attribute meaning to formulas of the language, for the logic system $(L, \vdash)$. Given a model $m \in \mathcal{M}$, the expression $m \vDash \delta$ will denote that the model m satisfies the formula δ. The tuple $(L, \mathcal{M}, \vDash)$ is called a **satisfaction system**.

Definition 2.2.3. Let $(L, \mathcal{M}, \vDash)$ be a satisfaction system and δ a formula. We say that:

(i) δ is **valid** (in $\mathcal{M}$), if $m \vDash \delta$ for all $m \in \mathcal{M}$;

(ii) δ is **satisfiable** (in $\mathcal{M}$), if exists $m \in \mathcal{M}$ such that $m \vDash \delta$;

(iii) δ is a **semantical consequence** (in $\mathcal{M}$) of the set of formulas Δ (denoted as $\Delta \vDash \delta$), if for all $m \in \mathcal{M}$ such that m satisfies each formula in Δ, then m satisfies δ as well.

The properties below in the next definition relate the concept of consequence operator $\vdash$ with the satisfaction relation $\vDash$, which have a major importance to any logic.

Definition 2.2.4. Let $\mathscr{L} = (L, \vdash)$ be a logic system and consider a semantics where $\mathcal{M}$ is the set of models and $\vDash$ is the satisfaction relation. We say that a logic system, relatively to the satisfaction system $(L, \mathcal{M}, \vDash)$, is:

(i) **strongly sound** if for every set of formulas Δ, any formula that is provable from Δ, follows semantically from Δ: if $\Delta \vdash \delta$ then $\Delta \vDash \delta$;

(ii) **weakly sound** if every provable formula is semantically valid: if $\vdash \delta$ then $\vDash \delta$;

(iii) **strongly complete** if for every set of formulas Δ, any formula that follows semantically from Δ, is provable from Δ: if $\Delta \vDash \delta$ then $\Delta \vdash \delta$;

(iv) **weakly complete** if every semantically valid formula is a theorem: if $\vDash \delta$ then $\vdash \delta$.

2.2.1 Hilbert Calculus

In general words, a Hilbert calculus is a formal deduction system characterized by a large number of axioms and a few number of inference rules (usually only *modus ponens*) opposing to natural deduction systems. The following definitions formalize this deduction system that will be used throughout this work.

Definition 2.2.5. A **Hilbert calculus** is a pair $\mathcal{H} = (L, R)$ where L is a set (of formulas) and $R = \{(\Delta_i, \delta_i) : i \in I\}$, where for each $i \in I$, $\Delta \subseteq L$ and $\delta \in L$.

Definition 2.2.6. Each pair $r = (\Delta, \delta) \in R$ is called an **inference rule**, and be denoted by $\{\delta_1, ..., \delta_n\} \vdash \delta$, where $\Delta = \{\delta_1, ..., \delta_n\}$. If $\Delta = \emptyset$, r is called an **axiom**. A **Hilbert calculus system** is a collection of axioms and inference rules.

Definition 2.2.7. Let $\mathcal{H} = (L, R)$ be a Hilbert calculus and $\Delta \subseteq L$ a set of formulas. We say that a formula $\delta \in L$ is **provable** (or **derivable**) from Δ (denoted as $\Delta \vdash \delta$) if there exists a finite sequence of formulas $\delta_1, ..., \delta_n \in L$ (called **proof**), where δ is δ_n and each formula δ_i $(i = 1, ..., n)$ is either in Δ or is the result of applying an inference rule $(\Delta_0, \delta_0) \in R$ such that $\Delta_0 \subseteq \{\delta_1, ..., \delta_{i-1}\}$ and δ_i is δ_0.

Moreover, we say that δ is a **theorem** if it is provable from $\emptyset$, and we write $\vdash \delta$.

We will write $\delta_1, ..., \delta_n \vdash_{\mathcal{H}} \delta$ instead of $\{\delta_1, ..., \delta_n\} \vdash_{\mathcal{H}} \delta$ to simplify the notation. It is easy to check that any segment of the proof of a provable formula is also a provable formula.

The most common notation to denote that δ is provable from Δ in $\mathcal{H} = (L, R)$ is $\Delta \vdash_{\mathcal{H}} \delta$. A Hilbert calculus $\mathcal{H} = (L, R)$ will always induce the logic system $(L, \vdash_{\mathcal{H}})$ (see Definition 2.2.1), that is, a Hilbert calculus is a particular case of an axiomatizable logic system.

2.2.2 Classical Propositional Logic

This subsection introduces Classical Propositional Logic (usually denoted as **CPL**) that will be used as a base logic over which will be further developed a probabilistic logic. In this work, this logic will be denoted as $\mathcal{B} = (B, \vdash_{\mathcal{B}})$. This is not the standard notation for CPL but, in this context, this notation makes sense because it will denote the **b**ase logic of the probabilistic system.

In the first place, the language (*i.e.* the syntax) of this logic system has to be described and some axioms and inference rules will be defined.

Definition 2.2.8. We will denote the set of formulas of CPL as $Form(\mathcal{B}) = \bigcup_{k=0}^{\infty} B_k$, where:

- $B_0 = Var(\mathcal{B}) = \{p, p_1, p_2, ...\}$
- $B_{n+1} = B_n \cup \{(\sim\phi) : \phi \in B_n\} \cup \{(\phi_1 \rightarrow \phi_2) : \phi_1, \phi_2 \in B_n\}$

In a logic context, this is described grammatically as follows.

$$\phi \ := \ p \mid (\sim\phi) \mid (\phi \rightarrow \phi) \ , \quad p \in Var(\mathcal{B})$$

Table 2.1: CPL Syntax

Note that it is only necessary to have in the language these two connectives ($\sim$ and $\to$), because any other classical connective can be written resorting to these two, as follows:

- $\top \ := \ \phi \to \phi$
- $\bot := \ \sim\!\top \ = \ \sim(\phi \to \phi)$
- $\phi_1 \vee \phi_2 \ := \ (\sim\phi_1) \to \phi_2$
- $\phi_1 \wedge \phi_2 \ := \sim\big((\sim\phi_1) \vee (\sim\phi_2)\big) \ = \ \sim\big(\phi_1 \to (\sim\phi_2)\big)$
- $\phi_1 \leftrightarrow \phi_2 \ := \ (\phi_1 \to \phi_2) \wedge (\phi_2 \to \phi_1) \ = \ \sim\big((\phi_1 \to \phi_2) \to \sim(\phi_2 \to \phi_1)\big)$

Table 2.2 gives an example of a Hilbert calculus for classical propositional logic. This inference system is present in [AR02].

Axioms:

[Ax1] $\vdash_{\mathcal{B}} \ \phi_1 \to (\phi_2 \to \phi_1)$

[Ax2] $\vdash_{\mathcal{B}} \ \big(\phi_1 \to (\phi_2 \to \phi_3)\big) \to \big((\phi_1 \to \phi_2) \to (\phi_1 \to \phi_3)\big)$

[Ax3] $\vdash_{\mathcal{B}} \ (\sim\phi_2 \to \sim\phi_1) \to \big((\sim\phi_2 \to \phi_1) \to \phi_2\big)$

Inference Rules:

[MP] $\phi_1, \phi_1 \to \phi_2 \ \vdash_{\mathcal{B}} \ \phi_2$

Table 2.2: Hilbert calculus for CPL

After having an inference system, our aim is now to introduce a semantics for this logic, *i.e.* to assign meaning to the symbols of this system. In CPL, meanings are truth values: 0 and 1 (false and true, respectively).

Definition 2.2.9. A **valuation** in CPL is a mapping $v : Var(\mathcal{B}) \to \{0,1\}$, that can be naturally extended to $\bar{v} : Form(\mathcal{B}) \to \{0,1\}$ by $\bar{v}(p) = v(p)$ if $p \in Var(\mathcal{B})$, $\bar{v}(\sim\phi) = 1 - \bar{v}(\phi)$ and $\bar{v}(\phi_1 \to \phi_2) = \max\{1 - \bar{v}(\phi_1), \bar{v}(\phi_2)\}$.

Usually, $\bar{v}$ is abbreviated to v because it is a more intuitive notation. In order to put these semantical definitions like Definition 2.2.2 of satisfaction system, we can see each valuation v as a model for CPL, and the satisfaction relation $\vDash_{\mathcal{B}}$ is recursively defined as:

- $v \vDash_{\mathcal{B}} p$ iff $v(p) = 1$;
- $v \vDash_{\mathcal{B}} (\sim\phi)$ iff $v \nvDash_{\mathcal{B}} \phi$;
- $v \vDash_{\mathcal{B}} (\phi_1 \to \phi_2)$ iff $(v \nvDash_{\mathcal{B}} \phi_1$ or $v \vDash_{\mathcal{B}} \phi_2)$.

The set $V := \big\{v \mid v : Var(\mathcal{B}) \to \{0,1\}\big\}$ will represent all models, *i.e.* all valuations, in CPL. Given $\phi \in Form(\mathcal{B})$, the subset $mod(\phi) := \{v \in V : v \vDash_{\mathcal{B}} \phi\}$ will denote all CPL models that satisfy the formula ϕ.

The concept of CPL tautology is also important because it will be usefull to define the propositional probabilistic logic in Chapter 3 (namely, its Hilbert calculus).

Definition 2.2.10. We say that $\phi \in Form(\mathcal{B})$ is a **tautology** if and only if $v(\phi) = 1$ for every valuation $v \in V$.

The following Theorem is very important for any logic system, and in the context of this work, it is necessary so that the base logic (CPL) of our probabilistic logic has such properties.

Theorem 2.2.11. CPL is strongly sound and complete.

In this work, the proof of this result will be not analyzed because our focus is the study of probabilistic logics (namely EPPL). Moreover, the soundness and completeness of CPL is studied in [Woj84] and [PW08]. Nonetheless, Theorem 2.2.11 has a major importance in the probabilistic logic studied in Chapter 3.

2.3 Real Closed Fields

In order to add probability concepts to CPL, first we need to define a logical structure to represent the real numbers over which probabilities will be defined. For that, we will follow the decidable first-order logic of *real closed fields* (**RCF**) over non-logical symbols $\{=, +, \times, <, 0, 1\}$, proposed in [Tar56] and simplified in [Bal10].

Next it will be define a first-order logic, to further introduce the RCF concepts.

Definition 2.3.1. Given a countable set $X = \{x_1, x_2, ...\}$ of variables, a family $F = (F_n)_{n>0}$ of function symbols and a family $R = (R_n)_{n>0}$ of relation symbols, we define a **first-order logic** with the language in Table 2.3 and the Hilbert calculus in Table 2.4.

$$t \ := \ x \mid f(t, ..., t) \ , \quad f \in F$$

$$\alpha \ := \ r(t, ..., t) \mid (\neg\alpha) \mid (\alpha \sqsupset \alpha) \mid (\forall x \alpha) \ , \quad r \in R$$

Table 2.3: First-Order Logic Syntax

Although there are two well-known first-order quantifiers for real numbers, the existential quantifier can be defined with recourse to the universal quantifier as usual: $(\exists x \phi) := \neg\big(\forall x (\neg\phi)\big)$.

Axioms:

[Ax1] $\vdash \ \forall x(\alpha_1 \sqsupset \alpha_2) \sqsupset (\forall x \alpha_1 \sqsupset \forall x \alpha_2)$

[Ax2] $\vdash \ (\alpha \sqsupset \forall x \alpha)$, if x is not free in α

[Ax3] $\vdash \ \big((\forall x \alpha) \sqsupset \phi[x \leftarrow t]\big)$, if t is free for x in α

Generalization inference rule:

[GIF] $\alpha \vdash \forall x \alpha$

Table 2.4: Hilbert calculus for First-Order Logic

Real Terms: $t \ := \ 0 \mid 1 \mid x \mid (t + t) \mid (t \times t) \ , \qquad x \in X$ First-Order Real Formulas: $\alpha \ := \ (t = t) \mid (t < t) \mid (\neg\alpha) \mid (\alpha \sqsupset \alpha) \mid (\forall x \alpha)$

Table 2.5: RCF first-order language

Then, given $X = \{x_1, x_2, ...\}$ a countable set (possibly infinite) of variables, the first-order language for RCF is given by:

Given an assignment to real variables $\rho : X \to \mathbb{R}$, the interpretation of real terms is defined as:

- $[0]_\rho = 0$; $[1]_\rho = 1$;
- $[x]_\rho = \rho(x)$;
- $[t_1 + t_2]_\rho = [t_1]_\rho + [t_2]_\rho$;
- $[t_1 \times t_2]_\rho = [t_1]_\rho \times [t_2]_\rho$;

A satisfaction relation $\vDash_{\mathrm{RFC}(\rho)}$ is defined by:

- $\vDash_{\mathrm{RFC}(\rho)} (t_1 = t_2)$ iff $[t_1]_\rho = [t_2]_\rho$;
- $\vDash_{\mathrm{RFC}(\rho)} (t_1 < t_2)$ iff $[t_1]_\rho < [t_2]_\rho$;
- $\vDash_{\mathrm{RFC}(\rho)} (\neg\alpha)$ iff $\nvDash_{\mathrm{RFC}(\rho)} \alpha$
- $\vDash_{\mathrm{RFC}(\rho)} (\alpha_1 \sqsupset \alpha_2)$ iff $(\nvDash_{\mathrm{RFC}(\rho)} \alpha_1$ or $\vDash_{\mathrm{RFC}(\rho)} \alpha_2)$;
- $\vDash_{\mathrm{RFC}(\rho)} (\forall x \alpha)$ iff $\vDash_{\mathrm{RFC}(\rho')} \alpha$ for all assignments ρ' such that ρ' agrees with ρ in the values of all variables different from x.

Together with the axiomatic systems of CPL (Table 2.2) and first-order logic (Table 2.4), we will define the remaining axioms necessary in order to have an axiomatic formal system for RCF (Table 2.6).

It is possible to prove that the axiomatic system of RCF is strongly sound and weakly complete. To end this section, we will make a remark about algebraic real numbers and first-order logic.

Definition 2.3.2. A real number $r \in \mathbb{R}$ is an **algebraic real number** if it is a root of a non-zero polynomial in one variable, with rational coefficients.

We will denote as $\mathrm{Alg}(\mathbb{R})$ the countable set of all algebraic real numbers. An important fact is that we can add algebraic real numbers to our logic because it is possible to prove that any $r \in \mathrm{Alg}(\mathbb{R})$ can be represented as a RCF formula (see [BPR03]).

For each $r \in \mathrm{Alg}(\mathbb{R})$, we will consider the RCF existential formula $\phi_r(x)$, with exactly one free variable $x \in X$ such that for each assignment $\rho : X \to \mathbb{R}$, we have that $\rho \vDash_{\mathrm{RCF}} \phi_r(x)$ if and only if $\rho(x) = r$.

Field Axioms:

[F1]	$\forall x \forall y \forall z(((x+y)+z) = (x+(y+z)))$
[F2]	$\forall x(x+0 = x)$
[F3]	$\forall x \exists y(x+y = 0)$
[F4]	$\forall x \forall y(x+y = y+x)$
[F5]	$\forall x \forall y \forall z(((x \times y) \times z) = (x \times (y \times z)))$
[F6]	$\forall x(x \times 1 = x)$
[F7]	$\forall x((x \neq 0) \sqsupset (\exists y(x \times y = 1)))$
[F8]	$\forall x \forall y(x \times y = y \times x)$
[F9]	$\forall x \forall y \forall z(x \times (y+z) = (x \times y) + (x \times z))$
[F10]	$(0 \neq 1)$

Linear Order Axioms:

[LO1]	$\forall x(\neg(x < x))$
[LO2]	$\forall x \forall y \forall z(((x < y) \sqcap (y < z)) \sqsupset (x < z))$
[LO3]	$\forall x \forall y((x < y) \sqcup (x = y) \sqcup (y < x))$

Ordered Fields Axioms:

[OF1]	$\forall x \forall y \forall z((x < y) \sqsupset ((x+z) < (y+z)))$
[OF1]	$\forall x \forall y(((0 < x) \sqcap (0 < y)) \sqsupset (0 < x \times y))$

Closed Fields Axioms:

[CF1]	$\forall x((0 < x) \sqsupset (\exists y(y \times y = x)))$
[CF2]	$\forall x_1 ... \forall x_{2n} \exists y(y^{2n+1} + \sum_{i=0}^{2n} x_i \times y^i = 0)$

Table 2.6: RCF Axiomatic System

Chapter 3

Probabilistic Propositional Logic

In the previous chapter, we defined a logical structure that we will use as a base (CPL) and a logical structure for real numbers (RCF). We are now able to formally define a probabilistic propositional logic.

In this chapter, we will follow some of the ideas in [MSS05], [BRS06], [BMNP07] and [BM09]. Taking into account the creators of this logic, we will keep, in this dissertation, its original designation - *Exogenous Probabilistic Propositional Logic* (EPPL).

The *exogenous* semantics approach to enriching a logic system (in this case, CPL) consists in defining each model in the enrichment as a set of models in the original logic plus some relevant structure, preserving both the syntax and semantics of the base logic.

In the following sections first it will be defined a syntax for EPPL, that has all the formulas from CPL as atoms (Section 3.1). Then it will be defined a semantics (Section 3.2) and a Hilbert calculus (Section 3.3). Some interessing results adapted from Probability Theory to EPPL will be proved. Finally, in this chapter will be analysed soundness and completeness for this probabilistic logic.

3.1 Syntax

In this section, we will define the syntax of this logic, by describing what does each symbol represent. The complete and minimal syntax of the EPPL is presented in Table 3.1.

Local Formulas:

$$\phi \ := \ p \mid (\sim\phi) \mid (\phi \rightarrow \phi) \ , \qquad p \in \mathit{Var}(\mathcal{B})$$

Real Probabilistic Terms:

$$t \ := \ x \mid 0 \mid 1 \mid \textstyle\int\phi \mid (t + t) \mid (t \times t) \ , \ \ x \in \mathit{Var}(\mathbb{R})$$

Global Formulas:

$$\delta \ := \ \Box\phi \mid (t \leq t) \mid (\neg\delta) \mid (\delta \sqsupset \delta) \ .$$

Table 3.1: EPPL Syntax

In very general terms, the language of EPPL consists of propositional formulas at two different levels: local and global formulas. The local ones are formulas from CPL and the global formulas allow reasoning about real probabilistic terms.

The real probabilistic terms ($t, t_1, t_2, ...$) denote elements of the algebraic real closed field, *i.e.* real numbers used for quantitative reasoning. The term $\int\phi$ denotes the probability of the events described by the local formula ϕ. In the literature, these terms are usually called *measure terms*. The global formulas ($\delta, \delta_1, \delta_2, ...$) are constituted by universal formulas $\Box\phi$, comparisons between terms ($t_1 \leq t_2$), and two connectives: global negation and global implication ($\neg$ and $\sqsupset$, respectively).

The global universal formula ($\Box\phi$) imposes that all elements of the probability space satisfy the local formula ϕ. In EPPL, the box symbol has to be interpretated not as a classical modality (*e.g.* $\Box(\Box\phi)$ is not a well defined formula), but instead as a unary symbol that allow a local formula to be interpretated in a global context. We can use ($\Diamond\phi$) to abbreviate $\neg\big(\Box(\sim\phi)\big)$, and this formula impose that there is at least one sample point that satisfies ϕ.

In a certain way, we can see these global formulas as propositional formulas where universal formulas and comparisons between terms are the atomic formulas of this logic. As it happens in CPL, there will be other connectives for global formulas which are introduced in the syntax as standard abbreviations from global negation and global implication:

- $\mathbf{T} \;:=\; \delta \sqsupset \delta$
- $\mathbf{F} \;:=\; \neg\mathbf{T}$
- $\delta_1 \sqcup \delta_2 \;:=\; (\neg\delta_1) \sqsupset \delta_2$
- $\delta_1 \sqcap \delta_2 \;:=\; \neg\big((\neg\delta_1) \sqcup (\neg\delta_2)\big)$
- $\delta_1 \equiv \delta_2 \;:=\; (\delta_1 \sqsupset \delta_2) \sqcap (\delta_2 \sqsupset \delta_1)$

Analogously, all notation for comparisons between real terms can be abbreviated as global formulas:

- $(t_1 = t_2) := (t_1 \leq t_2) \sqcap (t_2 \leq t_1)$
- $(t_1 \neq t_2) := \neg(t_1 = t_2)$
- $(t_1 > t_2) := \neg(t_1 \leq t_2)$
- $(t_1 \geq t_2) := (t_1 > t_2) \sqcup (t_1 = t_2)$
- $(t_1 < t_2) := \neg(t_1 \geq t_2)$

In the following sections of this work, this language will be developed, namely by defining its semantics (Section 3.2) and a Hilbert calculus (Section 3.3). In these sections some classical results of Probability Theory will be proved in the context of EPPL (both semantically and in the calculus defined) and, finally will be discussed both soundness (Section 3.4) and completeness (Section 3.5) for EPPL.

3.2 Semantics

Here we will define a satisfaction system for EPPL and then we will study some properties of this semantics that will be useful later to prove the soundness of EPPL. In order to define a model of EPPL, we will make use of the concepts of probability space and stochastic process previously defined.

Definition 3.2.1. Let $(\Omega, \mathcal{F}, \mathbf{P})$ be an arbitrary probability space. A **model** of EPPL is a tuple $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ where $\mathbf{X} = (X_p)_{p \in Var(\mathcal{B})}$ is a stochastic process over this probability space. Each X_p represents a Bernoulli random variable, *i.e.* $X_p : \Omega \to \{0, 1\}$.

The purpose of the next example is two illustrate how an EPPL model represents a probabilistic random experiment.

Example 3.2.2. Consider the random experiment of a dice roll. The EPPL model associated is $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$, where $\Omega = \{1, 2, 3, 4, 5, 6\}$ is the sample space, $\mathcal{F} = \mathcal{P}(\Omega)$ and $\mathbf{P}(\omega) = 1/6$ for each $\omega \in \Omega$ compose the probability space. $\mathbf{X} = (X_p)_{p \in Var(\mathcal{B})}$ is a stochastic process over this probability space, namely $X_p : \Omega \to \{0, 1\}$ is a Bernoulli random variable.

Suppose that, for example, p describes the event *even number occurs*. This Bernoulli r.v. associates each $p \in \mathrm{Var}(\mathcal{B})$ with 1 if and only if the sample point ω is consistent with the meaning of p. That is, $X_p(\omega) = 1$ iff ω is an even number ($\omega \in \{2, 4, 6\}$).

Note that every local formula ϕ will induce a new Bernoulli r.v. $X_\phi : \Omega \to \{0, 1\}$, defined recursively as follow:

$$X_{\sim\phi}(\omega) = 1 - X_\phi(\omega) \quad \text{and} \quad X_{\phi_1 \to \phi_2}(\omega) = \max\{1 - X_{\phi_1}(\omega), X_{\phi_2}(\omega)\}.$$

Then any local formula ϕ will be represented in the probability space as a subset of the sample set Ω, *i.e.* $\Omega_\phi = \{\omega \in \Omega : X_\phi(\omega) = 1\}$. We can intuitively observe that each sample point $\omega \in \Omega$ will induce a valuation $v_\omega(p)$ in the local CPL, as defined in 2.2.9, such that $v_\omega(p) = X_p(\omega)$, for all propositions $p \in Var(\mathcal{B})$.

Reciprocally, the local logic will induce a probability space $(\Omega, \mathcal{F}, \mathbf{P}) := \big(V, \mathcal{P}(V), \mathbf{P}\big)$ where the sample points are valuations of CPL with some probability $\mathbf{P}(\{v\})$ associated and each local formula ϕ will induce this time a Bernoulli r.v. such that $X_\phi(v) = v(\phi)$.

The next proposition shows how this Bernoulli random variables behaves according to the abbreviations defined in CPL (see Subsection 2.2.2).

Proposition 3.2.3. Let ϕ_1 and ϕ_2 be two local formulas and $\omega \in \Omega$ an arbitrary sample point. Considering the abbreviations defined for local formulas in Subsection 2.2.2, we have that:

(i) $X_\top(\omega) = 1$;

(ii) $X_\bot(\omega) = 0$;

(iii) $X_{\phi_1 \vee \phi_2}(\omega) = \max\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\}$;

(iv) $X_{\phi_1 \wedge \phi_2}(\omega) = \min\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\}$;

(v) $X_{\phi_1 \leftrightarrow \phi_2}(\omega) = \begin{cases} 1, & \text{if } X_{\phi_1}(\omega) = X_{\phi_2}(\omega) \\ 0, & \text{otherwise} \end{cases}$.

Proof.

(i) $X_\top(\omega) = X_{\phi \to \phi}(\omega) = \max\{1 - X_\phi(\omega), X_\phi(\omega)\} = 1$, because $X_\phi(\omega) \in \{0, 1\}$.

(ii) $X_{\perp}(\omega) = X_{\sim\top}(\omega) = 1 - X_{\top}(\omega) = 1 - 1 = 0.$

(iii) $X_{\phi_1 \vee \phi_2}(\omega) = X_{(\sim\phi_1)\to\phi_2}(\omega) = \max\{1 - X_{\sim\phi_1}(\omega), X_{\phi_2}(\omega)\}$
$= \max\big\{1 - \big(1 - X_{\phi_1}(\omega)\big), X_{\phi_2}(\omega)\big\} = \max\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\}.$

(iv) $X_{\phi_1 \wedge \phi_2}(\omega) = X_{\sim((\sim\phi_1)\vee(\sim\phi_2))}(\omega) = 1 - X_{(\sim\phi_1)\vee(\sim\phi_2)}(\omega) = 1 - \max\{X_{\sim\phi_1}(\omega), X_{\sim\phi_2}(\omega)\}$
$= 1 - \max\{1 - X_{\phi_1}(\omega), 1 - X_{\phi_2}(\omega)\} = \min\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\}.$

(v) $X_{\phi_1 \leftrightarrow \phi_2}(\omega) = X_{(\phi_1\to\phi_2)\wedge(\phi_2\to\phi_1)}(\omega) = \min\{X_{\phi_1\to\phi_2}(\omega), X_{\phi_2\to\phi_1}(\omega)\}$
$= \min\big\{\max\{1 - X_{\phi_1}(\omega), X_{\phi_2}(\omega)\},\ \max\{1 - X_{\phi_2}(\omega), X_{\phi_1}(\omega)\}\big\}.$
If $X_{\phi_1}(\omega) = X_{\phi_2}(\omega)$, then:
$X_{\phi_1 \leftrightarrow \phi_2}(\omega) = \min\big\{\max\{1 - X_{\phi_2}(\omega), X_{\phi_2}(\omega)\},\ \max\{1 - X_{\phi_2}(\omega), X_{\phi_2}(\omega)\}\big\}$
$= \max\{1 - X_{\phi_2}(\omega), X_{\phi_2}(\omega)\} = 1$, because $X_{\phi_2}(\omega) \in \{0, 1\}$;
On the other hand, if $X_{\phi_1}(\omega) = 1 - X_{\phi_2}(\omega)$, then:
$X_{\phi_1 \leftrightarrow \phi_2}(\omega) = \min\big\{\max\{X_{\phi_2}(\omega), X_{\phi_2}(\omega)\},\ \max\{1 - X_{\phi_2}(\omega), 1 - X_{\phi_2}(\omega)\}\big\}$
$= \min\{X_{\phi_2}(\omega), 1 - X_{\phi_2}(\omega)\} = 0$, because $X_{\phi_2}(\omega) \in \{0, 1\}$.

□

Now our aim is to define a satisfaction relation (see Definition 2.2.2) for global formulas. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model and $\rho : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$ an assignment to the real variables (*i.e.* an assignment of real variables to algebraic numbers). The interpretation of probabilistic real terms $[t]_{(m,\rho)}$ is as follows:

- $[0]_{(m,\rho)} := 0,\ [1]_{(m,\rho)} := 1;$
- $[x]_{(m,\rho)} := \rho(x)$, for all $x \in \mathit{Var}(\mathbb{R})$
- $[t_1 + t_2]_{(m,\rho)} := [t_1]_{(m,\rho)} + [t_2]_{(m,\rho)}$
- $[t_1 \times t_2]_{(m,\rho)} := [t_1]_{(m,\rho)} \times [t_2]_{(m,\rho)}$
- $[\int \phi]_{(m,\rho)} = \mathbf{P}(X_\phi = 1) = \mathbf{P}(\{\omega \in \Omega : X_\phi(\omega) = 1\}) = \mathbf{P}(\Omega_\phi)$

When a real term t does not have any real variable, we will abbreviate the notation as $[t]_m := [t]_{(m,\rho)}$, since the assignment ρ has no contribution to the final outcome. Note that this condition implies that $[r]_{(m,\rho)} := r$, for all $r \in \mathrm{Alg}(\mathbb{R})$.

The following proposition shows that there exists a duality between the CPL connectives and the connectives of set theory behind probability spaces. This results will be used later in this work, for example in order to prove EPPL weak soundness.

Proposition 3.2.4. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model and ϕ, ϕ_1, $\phi_2 \in \mathrm{Form}(\mathcal{B})$.

(i) $[\int(\sim\phi)]_m = \mathbf{P}(\Omega \backslash \Omega_\phi)$

(ii) $[\int(\phi_1 \vee \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1} \cup \Omega_{\phi_2})$

(iii) $[\int(\phi_1 \wedge \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2})$

Proof.

(i) $[\int(\sim\phi)]_m = \mathbf{P}(\Omega_{\sim\phi}) = \mathbf{P}(\{\omega \in \Omega : X_{\sim\phi}(\omega) = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : 1 - X_\phi(\omega) = 1\}) = \mathbf{P}(\{\omega \in \Omega : X_\phi(\omega) = 0\})$
$= \mathbf{P}(\Omega \backslash \{\omega \in \Omega : X_\phi(\omega) = 1\}) = \mathbf{P}(\Omega \backslash \Omega_\phi)$

(ii) $[\int(\phi_1 \vee \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1 \vee \phi_2}) = \mathbf{P}(\{\omega \in \Omega : X_{\phi_1 \vee \phi_2}(\omega) = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : \max\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\} = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : X_{\phi_1}(\omega) = 1 \text{ or } X_{\phi_2}(\omega) = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : X_{\phi_1}(\omega) = 1\} \cup \{\omega \in \Omega : X_{\phi_2}(\omega) = 1\}) = \mathbf{P}(\Omega_{\phi_1} \cup \Omega_{\phi_2})$

(iii) $[\int(\phi_1 \wedge \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2}) = \mathbf{P}(\{\omega \in \Omega : X_{\phi_1 \wedge \phi_2}(\omega) = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : \min\{X_{\phi_1}(\omega), X_{\phi_2}(\omega)\} = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : X_{\phi_1}(\omega) = 1 \text{ and } X_{\phi_2}(\omega) = 1\})$
$= \mathbf{P}(\{\omega \in \Omega : X_{\phi_1}(\omega) = 1\} \cap \{\omega \in \Omega : X_{\phi_2}(\omega) = 1\}) = \mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2})$

□

We can now define the satisfaction relation (see Definition 2.2.2) for our probabilistic logic, that is, define the satisfaction system for EPPL.

Definition 3.2.5. Given an EPPL model $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ and an assignment to the real variables $\rho : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$, the **satisfaction** of a global formula δ, denoted as $(m, \rho) \vDash \delta$, is recursively defined as follow:

- $(m, \rho) \vDash \Box\phi$ iff $(X_\phi(\omega) = 1 \text{ for all } \omega \in \Omega)$ iff $(\Omega_\phi = \Omega)$
- $(m, \rho) \vDash (t_1 \leq t_2)$ iff $[t_1]_{(m,\rho)} \leq [t_2]_{(m,\rho)}$
- $(m, \rho) \vDash (\neg\delta)$ iff $(m, \rho) \nvDash \delta$
- $(m, \rho) \vDash (\delta_1 \sqsupset \delta_2)$ iff $\big((m, \rho) \nvDash \delta_1 \text{ or } (m, \rho) \vDash \delta_2\big)$

If $(m, \rho) \vDash \delta$ for every real assignment ρ, then we will abbreviate the notation to $m \vDash \delta$. Moreover, if $m \vDash \delta$ for every EPPL model m, we simply write $\vDash \delta$, and δ is called a **valid** EPPL formula.

The next proposition shows how global connective defined as abbreviations behave concerning to this satisfaction relation.

Proposition 3.2.6. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model and $\rho : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$ an assignment to the real variables. Given two EPPL formula δ_1 and δ_2, and considering the abbreviations previously defined, we have that:

(i) $(m, \rho) \vDash \mathbf{T}$ always ;

(ii) $(m, \rho) \vDash \mathbf{F}$ never ;

(iii) $(m, \rho) \vDash (\delta_1 \sqcup \delta_2)$ iff $\big((m, \rho) \vDash \delta_1$ or $(m, \rho) \vDash \delta_2\big)$;

(iv) $(m, \rho) \vDash (\delta_1 \sqcap \delta_2)$ iff $\big((m, \rho) \vDash \delta_1$ and $(m, \rho) \vDash \delta_2\big)$;

(v) $(m, \rho) \vDash (\delta_1 \equiv \delta_2)$ iff $\big((m, \rho) \nvDash \delta_1$ and $(m, \rho) \nvDash \delta_2\big)$ or $\big((m, \rho) \vDash \delta_1$ and $(m, \rho) \vDash \delta_2\big)$.

Proof.

(i) $(m, \rho) \vDash \mathbf{T}$ iff $(m, \rho) \vDash (\delta_1 \sqsupset \delta_1)$ iff $\big((m, \rho) \nvDash \delta_1$ or $(m, \rho) \vDash \delta_1\big)$, which is always true because these two statements are complementary.

(ii) $(m, \rho) \vDash \mathbf{F}$ iff $(m, \rho) \vDash \neg\mathbf{T}$ iff $(m, \rho) \nvDash \mathbf{T}$, which is always false by (i).

(iii) $(m, \rho) \vDash (\delta_1 \sqcup \delta_2)$ iff $(m, \rho) \vDash (\neg\delta_1 \sqsupset \delta_2)$ iff $\big((m, \rho) \nvDash \neg\delta_1$ or $(m, \rho) \vDash \delta_2\big)$
iff $\big((m, \rho) \vDash \delta_1$ or $(m, \rho) \vDash \delta_2\big)$.

(iv) $(m, \rho) \vDash (\delta_1 \sqcap \delta_2)$ iff $(m, \rho) \vDash \neg(\neg\delta_1 \sqcup \neg\delta_2)$ iff $(m, \rho) \nvDash (\neg\delta_1 \sqcup \neg\delta_2)$
iff is false that $\{(m, \rho) \vDash (\neg\delta_1)$ or $(m, \rho) \vDash (\neg\delta_2)\}$
iff $\big((m, \rho) \nvDash (\neg\delta_1)$ and $(m, \rho) \nvDash (\neg\delta_2)\big)$ iff $\big((m, \rho) \vDash \delta_1$ and $(m, \rho) \vDash \delta_2\big)$.

(v) $(m, \rho) \vDash (\delta_1 \equiv \delta_2)$ iff $(m, \rho) \vDash (\delta_1 \sqsupset \delta_2) \sqcap (\delta_2 \sqsupset \delta_1)$
iff $\big((m, \rho) \vDash (\delta_1 \sqsupset \delta_2)$ and $(m, \rho) \vDash (\delta_2 \sqsupset \delta_1)\big)$
iff $\big((m, \rho) \nvDash \delta_1$ or $(m, \rho) \vDash \delta_2\big)$ and $\big((m, \rho) \nvDash \delta_2$ or $(m, \rho) \vDash \delta_1\big)$
iff $\big((m, \rho) \nvDash \delta_1$ and $(m, \rho) \nvDash \delta_2\big)$ or $\big((m, \rho) \nvDash \delta_1$ and $(m, \rho) \vDash \delta_1\big)$
or $\big((m, \rho) \vDash \delta_2$ and $(m, \rho) \nvDash \delta_2\big)$ or $\big((m, \rho) \vDash \delta_2$ and $(m, \rho) \vDash \delta_1\big)$
iff $\big((m, \rho) \nvDash \delta_1$ and $(m, \rho) \nvDash \delta_2\big)$ or $\big((m, \rho) \vDash \delta_1$ and $(m, \rho) \vDash \delta_2\big)$.

□

Proposition 3.2.7. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model and $\rho : \text{Var}(\mathbb{R}) \to \text{Alg}(\mathbb{R})$ an assignment to the real variables. Given two real probabilistic terms t_1 and t_2, and considering the abbreviations previously defined, we have that:

(i) $(m, \rho) \vDash (t_1 = t_2)$ iff $[t_1]_{(m,\rho)} = [t_2]_{(m,\rho)}$;

(ii) $(m, \rho) \vDash (t_1 \neq t_2)$ iff $[t_1]_{(m,\rho)} \neq [t_2]_{(m,\rho)}$;

(iii) $(m, \rho) \vDash (t_1 > t_2)$ iff $[t_1]_{(m,\rho)} > [t_2]_{(m,\rho)}$;

(iv) $(m, \rho) \vDash (t_1 \geq t_2)$ iff $[t_1]_{(m,\rho)} \geq [t_2]_{(m,\rho)}$;

(v) $(m, \rho) \vDash (t_1 < t_2)$ iff $[t_1]_{(m,\rho)} < [t_2]_{(m,\rho)}$.

Proof.

(i) $(m, \rho) \vDash (t_1 = t_2)$ iff $(m, \rho) \vDash (t_1 \leq t_2) \sqcap (t_2 \leq t_1)$
iff $\big((m, \rho) \vDash (t_1 \leq t_2)\big)$ and $\big((m, \rho) \vDash (t_2 \leq t_1)\big)$
iff $([t_1]_{(m,\rho)} \leq [t_2]_{(m,\rho)})$ and $([t_2]_{(m,\rho)} \leq [t_1]_{(m,\rho)})$ iff $[t_1]_{(m,\rho)} = [t_2]_{(m,\rho)}$.

(ii) $(m, \rho) \vDash (t_1 \neq t_2)$ iff $(m, \rho) \vDash \neg(t_1 = t_2)$ iff $(m, \rho) \nvDash (t_1 = t_2)$

iff is false that $([t_1]_{(m,\rho)} = [t_2]_{(m,\rho)})$ iff $[t_1]_{(m,\rho)} \neq [t_2]_{(m,\rho)}$.

(iii) $(m,\rho) \vDash (t_1 > t_2)$ iff $(m,\rho) \vDash \neg(t_1 \leq t_2)$ iff $(m,\rho) \nvDash (t_1 \leq t_2)$
iff is false that $([t_1]_{(m,\rho)} \leq [t_2]_{(m,\rho)})$ iff $[t_1]_{(m,\rho)} > [t_2]_{(m,\rho)}$.

(iv) $(m,\rho) \vDash (t_1 \geq t_2)$ iff $(m,\rho) \vDash (t_1 > t_2) \sqcup (t_1 = t_2)$
iff $\big((m,\rho) \vDash (t_1 > t_2)\big)$ or $\big((m,\rho) \vDash (t_1 = t_2)\big)$
iff $([t_1]_{(m,\rho)} > [t_2]_{(m,\rho)})$ or $([t_1]_{(m,\rho)} = [t_2]_{(m,\rho)})$ iff $[t_1]_{(m,\rho)} \geq [t_2]_{(m,\rho)}$.

(v) $(m,\rho) \vDash (t_1 < t_2)$ iff $(m,\rho) \vDash \neg(t_1 \geq t_2)$ iff $(m,\rho) \nvDash (t_1 \geq t_2)$
iff is false that $([t_1]_{(m,\rho)} \geq [t_2]_{(m,\rho)})$ iff $[t_1]_{(m,\rho)} < [t_2]_{(m,\rho)}$.

□

Proposition 3.2.8. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model and $\rho_1 : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$ and $\rho_2 : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$ two assignments to the real variables. If δ is an EPPL formula with no occurrences of real variables, then $(m,\rho_1) \vDash \delta$ if and only if $(m,\rho_2) \vDash \delta$.

In this case, we will simplify the notation by omitting ρ. Note that, in general, we do not have that $m \vDash \Box(p_1 \vee p_2)$ (because $(p_1 \vee p_2)$ is not a CPL tautology), but in some EPPL models (as in the following example) is a valid formula.

Example 3.2.9. Consider the EPPL model in Example 3.2.2. Suppose that p_1 describes the event *even number occurs*, and p_2 describes the event *odd number occurs*. We have that $m \vDash \Box(p_1 \vee p_2)$, because $\Omega_{(p_1 \vee p_2)} = \Omega_{p_1} \cup \Omega_{p_2} = \{2,4,6\} \cup \{1,3,5\} = \{1,2,3,4,5,6\} = \Omega$.

The following results are two of the most popular results known in Probability Theory. We may clearly see the connection between Probability Theory and the probabilistic propositional logic defined here.

Proposition 3.2.10. $\vDash \int(\sim\phi) + \int\phi = 1$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. First note that $\Omega_\phi = \{\omega \in \Omega : X_\phi(\omega) = 1\}$ and $\Omega_{\sim\phi} = \{\omega \in \Omega : X_{\sim\phi}(\omega) = 1\} = \{\omega \in \Omega : 1 - X_\phi(\omega) = 1\} = \{\omega \in \Omega : X_\phi(\omega) = 0\}$ are two complementary sets over Ω. Then, $\mathbf{P}(\Omega) = \mathbf{P}(\Omega_{\sim\phi} \cup \Omega_\phi) = \mathbf{P}(\Omega_{\sim\phi}) + \mathbf{P}(\Omega_\phi)$. And, by definition of probability function, $\mathbf{P}(\Omega) = 1$. Then $\mathbf{P}(\Omega_{\sim\phi}) + \mathbf{P}(\Omega_\phi) = 1$ and:
$\mathbf{P}(\Omega_{\sim\phi}) + \mathbf{P}(\Omega_\phi) = 1$ iff $[\int(\sim\phi)]_m + [\int\phi]_m = [1]_m$ iff $[\int(\sim\phi) + \int\phi]_m = [1]_m$
iff $m \vDash \int(\sim\phi) + \int\phi = 1$.

□

Proposition 3.2.11. $\vDash \int\phi_1 + \int\phi_2 = \int(\phi_1 \vee \phi_2) + \int(\phi_1 \wedge \phi_2)$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. By Proposition 3.2.4, we have that:
$[\int\phi_1]_m = \mathbf{P}(\Omega_{\phi_1}) = \mathbf{P}(\{\omega \in \Omega : X_{\phi_1}(\omega) = 1\})$;
$[\int\phi_2]_m = \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\{\omega \in \Omega : X_{\phi_2}(\omega) = 1\})$;

$[\int(\phi_1 \vee \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1 \vee \phi_2}) = \mathbf{P}(\Omega_{\phi_1} \cup \Omega_{\phi_2})$;
$[\int(\phi_1 \wedge \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2}) = \mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2})$.

(i) $\Omega_{\phi_1 \vee \phi_2} = \Omega_{\phi_1} \cup \Omega_{\phi_2} = \Omega \cap (\Omega_{\phi_1} \cup \Omega_{\phi_2}) \quad = (\Omega_{\sim\phi_1} \cup \Omega_{\phi_1}) \cap (\Omega_{\phi_1} \cup \Omega_{\phi_2})$
$= (\Omega_{\sim\phi_1} \cap \Omega_{\phi_1}) \cup (\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup (\Omega_{\phi_1} \cap \Omega_{\phi_1}) \cup (\Omega_{\phi_1} \cap \Omega_{\phi_2})$
$= \emptyset \cup (\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup \Omega_{\phi_1} \cup (\Omega_{\phi_1} \cap \Omega_{\phi_2})$
$= (\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup \Omega_{\phi_1}$, because $\Omega_{\phi_1} \cap \Omega_{\phi_2} \subseteq \Omega_{\phi_1}$.
Then, $\mathbf{P}(\Omega_{\phi_1 \vee \phi_2}) = \mathbf{P}\big((\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup \Omega_{\phi_1}\big) = \mathbf{P}(\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) + \mathbf{P}(\Omega_{\phi_1})$, because $\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}$ and Ω_{ϕ_1} are two disjoint sets.

(ii) $\Omega_{\phi_2} = (\Omega_{\sim\phi_1} \cup \Omega_{\phi_1}) \cap \Omega_{\phi_2} = (\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup (\Omega_{\phi_1} \cap \Omega_{\phi_2}) = (\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup \Omega_{\phi_1 \wedge \phi_2}$
Then, $\mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}((\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) \cup \Omega_{\phi_1 \wedge \phi_2}) = \mathbf{P}(\Omega_{\sim\phi_1} \cap \Omega_{\phi_2}) + \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2})$, since $\Omega_{\sim\phi_1}$ and $\Omega_{\phi_1 \wedge \phi_2} = \Omega_{\phi_1} \cap \Omega_{\phi_2}$ are two disjoint sets.

Therefore, putting together the equations (i) and (ii), we have that:
$\mathbf{P}(\Omega_{\phi_1}) + \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1 \vee \phi_2}) + \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2})$
iff $[\int\phi_1]_m + [\int\phi_2]_m = [\int(\phi_1 \vee \phi_2)]_m + [\int(\phi_1 \wedge \phi_2)]_m$
iff $[\int\phi_1 + \int\phi_2]_m = [\int(\phi_1 \vee \phi_2) + \int(\phi_1 \wedge \phi_2)]_m$
iff $m \vDash \int\phi_1 + \int\phi_2 = \int(\phi_1 \vee \phi_2) + \int(\phi_1 \wedge \phi_2)$.

□

In our language, there are set two formulas that can represent the logical equivalent to the certain event: $\Box\phi$ and $(\int\phi = 1)$. The next proposition shows that one of these formulas is stronger than the other (that is, the later is a semantical consequence of the first).

Proposition 3.2.12. Let ϕ be a local formula such that $\vDash (\Box\phi)$. Then $\vDash (\int\phi = 1)$.

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model, and suppose that $m \vDash (\Box\phi)$. By definition, we have that $m \vDash (\Box\phi)$ if and only if $\Omega_\phi = \Omega$. Thus, $\mathbf{P}(\Omega_\phi) = \mathbf{P}(\Omega) = 1$. Since $[\int\phi]_m = \mathbf{P}(\Omega_\phi)$ and $[1]_m := 1$, we conclude that $[\int\phi]_m = [1]_m$, which is equivalent to $m \vDash (\int\phi = 1)$.

□

Note that the reciprocal (of the result in Proposition 3.2.12) is not true in general because of problems with null probabilities. The next example easily shows that fact.

Example 3.2.13. Consider a coin tossing in which the coin is handled to always give heads. This experiment can be described by the following probability space:
$\Omega = \{H, T\}$, $\mathcal{F} = \mathcal{P}(\Omega) = \{\emptyset, \{H\}, \{T\}, \Omega\}$, $\mathbf{P}(\{H\}) = 1$ and $\mathbf{P}(\{T\}) = 0$.
The propositional local formula p will represent the statement *coin flip gives heads*. Then, we can associate the respective Bernoulli random variable $X_p : \Omega \to \{0, 1\}$ such that $X_p(H) = 1$ and $X_p(T) = 0$. Given m the EPPL model described by this probability space, we have that $m \vDash (\int p = 1)$ because $\mathbf{P}(\Omega_p) = \mathbf{P}(\{H\}) = 1$, but $m \nvDash (\Box p)$ because $X_p(T) \neq 1$.

However, as it is showed in the next proposition, under some conditions, the equivalence result holds in the semantical contex of EPPL.

Proposition 3.2.14. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model such that $\mathbf{P}(\{\omega\}) > 0$ for each sample-point $\omega \in \Omega$. Then, for each local formula ϕ, $m \vDash (\Box\phi)$ if and only if $m \vDash (\int \phi = 1)$.

Proof.
($\Rightarrow$) Follows from Proposition 3.2.12.
($\Leftarrow$) Suppose now that $m \vDash (\int \phi = 1)$, which is equivalent to $\mathbf{P}(\Omega_\phi) = 1$. We want to prove that $m \vDash (\Box\phi)$, *i.e.*, that $\Omega_\phi = \Omega$. Consider (by *reductio ad absurdum*) that $\Omega_\phi \neq \Omega$. Then exists $\omega_0 \in \Omega$ such that $\omega_0 \notin \Omega_\phi$, and $\mathbf{P}(\Omega_\phi \cup \{\omega_0\}) = \mathbf{P}(\Omega_\phi) + \mathbf{P}(\{\omega_0\}) = 1 + \mathbf{P}(\{\omega_0\}) > 1$, because $\mathbf{P}(\{\omega_0\}) > 0$, which is a contradiction to the definition of probability space. Therefore, $\Omega_\phi = \Omega$.

□

In [BMN10], it is assumed that the probability space does not have impossible events. Here we do not assume that because our aim is to work with a logic as close as possible to Probability Theory. Besides, under the conditions of Proposition 3.2.14, the box symbol could be removed from the syntax of EPPL because we could see it as an abbreviation of $(\int \phi = 1)$ (as it was in [MSS05]).

The following proposition aims to show that the global true formula $\mathbf{T}$ has a similar behavior to the globalization of the local true formula $\Box\top$. An analogous for the false formulas. Further, in the weak soundness theorem (see Theorem 3.4.1), it is proved that in fact they are equivalent formulas.

Proposition 3.2.15. Let δ be an arbitrary EPPL formula. Then:

(i) $\vDash \ \delta \sqsupset (\Box\top)$

(ii) $\vDash \ \delta \sqsupset \mathbf{T}$

(iii) $\vDash \ (\Box\bot) \sqsupset \delta$

(iv) $\vDash \ \mathbf{F} \sqsupset \delta$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an arbitrary EPPL model and $\rho : \mathrm{Var}(\mathbb{R}) \to \mathrm{Alg}(\mathbb{R})$.

(i) $(m, \rho) \vDash \delta \sqsupset (\Box\top)$ iff $\big((m, \rho) \nvDash \delta\big)$ or $\big((m, \rho) \vDash \Box\top\big)$ iff $\big((m, \rho) \nvDash \delta\big)$ or $\big(X_\top(\omega) = 1$ for all $\omega \in \Omega\big)$. $X_\top(\omega) = 1$ is proved proved in Proposition 3.2.3 (i). Thus, the result holds.

(ii) $(m, \rho) \vDash \delta \sqsupset \mathbf{T}$ iff $\big((m, \rho) \nvDash \delta\big)$ or $\big((m, \rho) \vDash \mathbf{T}\big)$. And $(m, \rho) \vDash \mathbf{T}$ is already proved in Proposition 3.2.6 (i).

(iii) $(m, \rho) \vDash (\Box\bot) \sqsupset \delta$ iff $\big((m, \rho) \nvDash \Box\bot\big)$ or $\big((m, \rho) \vDash \Box\delta\big)$ iff $\big((m, \rho) \vDash \Box\delta\big)$ or is false that $\big((m, \rho) \vDash \Box\bot\big)$. By Proposition 3.2.3 (ii), $X_\bot(\omega) = 0$ for each $\omega \in \Omega$. Then, $\Omega_\bot = \{\omega \in \Omega : X_\bot(\omega) = 1\} = \emptyset$ and it is proved that it is false that $(m, \rho) \vDash \Box\bot$.

(iv) $(m, \rho) \vDash \mathbf{F} \sqsupset \delta$ iff $\big((m, \rho) \nvDash \mathbf{F}\big)$ or $\big((m, \rho) \vDash \delta\big)$. And $(m, \rho) \nvDash \mathbf{F}$ is already proved in Proposition 3.2.6 (ii).

□

3.3 Hilbert Calculus

Following the work in [MSS05], [BRS06], [BMNP07] and [BM09], we adopted, in the context of this thesis, a Hilbert calculus for this logic, that is represented in Table 3.3. This is the one which has the most elementary axioms and, starting from these, we will prove some basic properties of Probability Theory.

First, it will be defined the concept of a global tautological formula which is a kind of globalization of a CPL tautology (note that $\mathscr{L}$ designate EPPL system).

Definition 3.3.1. We say that an EPPL formula δ is a (global) **tautological formula** if there exists a local tautology ϕ and a mapping $f : Var(\mathcal{B}) \to Form(\mathscr{L})$ such that $\delta = \bar{f}(\phi)$, where $\bar{f}$ is the naturaly extension of f to $\bar{f} : Form(\mathcal{B}) \to Form(\mathscr{L})$, defined recursively as follow: $\bar{f}(\sim\phi_1) = \neg\bar{f}(\phi_1)$ and $\bar{f}(\phi_1 \to \phi_2) = \bar{f}(\phi_1) \sqsupset \bar{f}(\phi_2)$, for all $\phi_1, \phi_2 \in Form(\mathcal{B})$.

The following table presents the Hilbert calculus for EPPL that we will use in this work, consisting of axioms for tautologies, axioms for universal formulas and global connectives, axioms for comparison between real terms, probabilistic Kolmogorov axioms and the *Modus Ponens* inference rule.

Axioms:

[LTaut]	$\vdash \Box\phi$, if ϕ is tautology in CPL
[GTaut]	$\vdash \delta$, if δ is a tautological formula
[Imp]	$\vdash \Box(\phi_1 \to \phi_2) \sqsupset (\Box\phi_1 \sqsupset \Box\phi_2)$
[EqF]	$\vdash (\Box\bot) \equiv \mathbf{F}$
[Kol1]	$\vdash \int\top = 1$
[Kol2]	$\vdash \big(\int\sim(\phi_1 \wedge \phi_2) = 1\big) \sqsupset \big(\int(\phi_1 \vee \phi_2) = (\int\phi_1) + (\int\phi_2)\big)$
[Kol3]	$\vdash \Box(\phi_1 \to \phi_2) \sqsupset (\int\phi_1 \leq \int\phi_2)$
[Real]	$\vdash (t_1 \leq t_2)$, for each valid analytical real inequality

Inference Rule:

[GMP]	$\delta_1, (\delta_1 \sqsupset \delta_2) \vdash \delta_2$

Table 3.2: Hilbert Calculus for EPPL

The first axiom (**[LTaut]**) allows to bring into this calculus every tautology of our local logic, CPL. Analogously, **[GTaut]** says that all global tautological formulas are theorems.

The next two axioms, **[Imp]** and **[EqF]**, are intended to connect the local formulas with the global ones. They are truly necessary to prove the completeness of this logic.

The Kolmogorov axioms (**[Kol1]**,**[Kol2]** and **[Kol3]**), also known as the probability axioms, bringing into the EPPL calculus all the properties of probabilities. These axioms can be seen originally in [Kol33], and here are adapted to EPPL the probabilistic axioms written in current language in [Ada98]. The axiom **[Real]** is also an important one because this axiom inserts in the calculus all properties of the Real Closed Fields.

In this type of deduction system, we only need the *Modus Ponens* inference rule because, together with the axioms, we can prove the other inference rules. The concepts of theorem, proof, and others, are as defined in Chapter 2. Note that the formulas in the calculus denote generic formulas that may be replaced by any local formula (in ϕ, ϕ_1 and ϕ_2) or global formula (in δ, δ_1 and δ_2). To show how this system works, we will now derive *Modus Tollens* rule.

Example 3.3.2. We can prove using only the axiom **[GTaut]** and the primary inference rule **[GMP]**, the derivate inference rule, *Modus Tollens*:

[GMT] $\delta_1 \sqsupset \delta_2 \;\vdash\; (\neg\delta_2) \sqsupset (\neg\delta_1)$

1.	$\delta_1 \sqsupset \delta_2$	**[Hyp]**
2.	$(\delta_1 \sqsupset \delta_2) \sqsupset \big((\neg\delta_2) \sqsupset (\neg\delta_1)\big)$	**[GTaut]**
3.	$(\neg\delta_2) \sqsupset (\neg\delta_1)$	**[GMP(2,1)]**

The next theorem is the key to all of the results in this section because it connects the concepts of provable formulas and global implication. The first theorem that we will prove is a simplified version of the Deduction Theorem (the complete proof of this theorem for CPL can be seen in [AR02]).

Theorem 3.3.3. Let δ_1 and δ_2 be arbitrary global formulas. Then,

$$\delta_1 \vdash \delta_2 \quad \text{if and only if} \quad \vdash \delta_1 \sqsupset \delta_2 \ .$$

Proof.

($\Rightarrow$) Suppose that $\delta_1 \vdash \delta_2$, *i.e.* there exists a finite proof (with n steps) of δ_2 from $\{\delta_1\}$. This can be proved using induction over the length of the proof as in classic calculus.

Base step: if $n = 1$ (the proof has just one step), then by definition of proof, δ_2 can be just one of two thing: δ_2 is an axiom or δ_2 is δ_1.

(i) If δ_2 is an axiom:

1.	$\vdash\ \delta_2$	**[Axiom]**
2.	$\vdash\ \delta_2 \sqsupset (\delta_1 \sqsupset \delta_2)$	**[GTaut]**
3.	$\vdash\ \delta_1 \sqsupset \delta_2$	**[GMP(1,2)]**

(ii) If δ_2 is δ_1:

1.	$\vdash\ \delta_1 \sqsupset \delta_1$	**[GTaut]**
2.	$\vdash\ \delta_1 \sqsupset \delta_2$	**[Subs(1)]**

Induction step: assume that whenever there is a proof with length $n \leq k$ of δ_2 from $\{\delta_1\}$, there is a proof of $\vdash \delta_1 \sqsupset \delta_2$. Now suppose that there is a proof with $n = k+1$ steps of δ_2 from $\{\delta_1\}$. Because our system only has one inference rule by definition, the last step is always either an application of this rule, an axiom or δ_1. The last two are demonstrated in the base step. Without loss of generality, if the last step is a consequence of the primate rule **[GMP]**, the proof can be generically represented by:

1.	δ_1	**[Hyp]**
	...	
i .	δ_3	
	...	
k .	$\delta_3 \sqsupset \delta_2$	
(k+1).	δ_2	**[GMP(i,k)]**

By induction hypothesis, because there is a proof with length k of $(\delta_3 \sqsupset \delta_2)$ from $\{\delta_1\}$, there is a proof of $\vdash \delta_1 \sqsupset (\delta_3 \sqsupset \delta_2)$. And, because there is a proof with length $i < k$ of δ_3 from $\{\delta_1\}$, there is a proof of $\vdash \delta_1 \sqsupset \delta_3$. Then:

1.	$\vdash \delta_1 \sqsupset (\delta_3 \sqsupset \delta_2)$	**[Ind-Hyp]**
2.	$\vdash \delta_1 \sqsupset \delta_3$	**[Ind-Hyp]**
3.	$\vdash \big(\delta_1 \sqsupset (\delta_3 \sqsupset \delta_2)\big) \sqsupset \big((\delta_1 \sqsupset \delta_3) \sqsupset (\delta_1 \sqsupset \delta_2)\big)$	**[GTaut]**
4.	$\vdash (\delta_1 \sqsupset \delta_3) \sqsupset (\delta_1 \sqsupset \delta_2)$	**[GMP(1,3)]**
5.	$\vdash \delta_1 \sqsupset \delta_2$	**[GMP(2,5)]**

($\Leftarrow$) Suppose now that $\vdash \delta_1 \sqsupset \delta_2$. We have to prove that $\delta_1 \vdash \delta_2$. This proof is trivial:

1.	$\delta_1 \sqsupset \delta_2$	**[Supp]**
2.	δ_1	**[Hyp]**
3.	δ_2	**[GMP(2,1)]**

□

In other words, this theorem says that $\vdash \delta_1 \sqsupset \delta_2$ if and only if δ_2 is provable from $\{\delta_1\}$. The next theorem is very useful because it allows us to split a global conjunction in its two components, and vice versa. Here, we get two new derived inference rules that we are going to apply on the next results.

Theorem 3.3.4. Let δ_1 and δ_2 be arbitrary global formulas. Then,
[GCi] $(\delta_1 \sqcap \delta_2) \vdash \delta_1$; $(\delta_1 \sqcap \delta_2) \vdash \delta_2$
[GCii] $\delta_1, \delta_2 \vdash (\delta_1 \sqcap \delta_2)$

Proof.

[GCi] Proof of δ_1 and δ_2 from $\{\delta_1 \sqcap \delta_2\}$:

1.	$\delta_1 \sqcap \delta_2$	**[Hyp]**
2.	$(\delta_1 \sqcap \delta_2) \sqsupset \delta_1$	**[GTaut]**
3.	$(\delta_1 \sqcap \delta_2) \sqsupset \delta_2$	**[GTaut]**
4.	δ_1	**[GMP(1,2)]**
5.	δ_2	**[GMP(1,3)]**

[GCii] Proof of $\delta_1 \sqcap \delta_2$ from $\{\delta_1, \delta_2\}$:

1.	δ_1	**[Hyp]**
2.	δ_2	**[Hyp]**
3.	$\delta_1 \sqsupset \big(\delta_2 \sqsupset (\delta_1 \sqcap \delta_2)\big)$	**[GTaut]**
4.	$\delta_2 \sqsupset (\delta_1 \sqcap \delta_2)$	**[GMP(1,3)]**
5.	$\delta_1 \sqcap \delta_2$	**[GMP(2,4)]**

□

The next derived inference rule that we will demonstrate is a version of *Modus Ponens*, this time applied only to local formulas but in a global context.

Proposition 3.3.5. **[LMP]** $\Box\phi_1, \Box(\phi_1 \to \phi_2) \vdash \Box\phi_2$

Proof. We have the following proof of $\Box\phi_2$ from $\{\Box\phi_1, \Box(\phi_1 \to \phi_2)\}$:

1.	$\Box\phi_1$	**[Hyp]**
2.	$\Box(\phi_1 \to \phi_2)$	**[Hyp]**
3.	$\Box(\phi_1 \to \phi_2) \sqsupset (\Box\phi_1 \sqsupset \Box\phi_2)$	**[Imp]**
4.	$\Box\phi_1 \sqsupset \Box\phi_2$	**[GMP(2,3)]**
5.	$\Box\phi_2$	**[GMP(1,4)]**

□

In some articles, this derived inference rule belongs to the calculus system but, because we can prove it from the system in Table 3.3, this local *Modus Ponens* rule is redundant, hence does not need to be considered as a primary rule (as is, for example, in [MSS05] and [BMNP07]).

In the next proposition it is shown that the global formulas $\Box(\phi_1 \wedge \phi_2)$ and $(\Box\phi_1) \sqcap (\Box\phi_2)$ are equivalent.

Proposition 3.3.6. **[Conj]** $\vdash \Box(\phi_1 \wedge \phi_2) \equiv \big((\Box\phi_1) \sqcap (\Box\phi_2)\big)$

Proof. In order to prove this global equivalence, considering that this equivalence is a syntactical abbreviation of a global conjunction of two implications, and also considering Theorem 3.3.4, we need to prove both global implications. And, because the result previously showed in Theorem 3.3.3, we can prove the following:

(i) Prove that $\Box(\phi_1 \wedge \phi_2) \vdash \big((\Box\phi_1) \sqcap (\Box\phi_2)\big)$:

1.	$\Box(\phi_1 \wedge \phi_2)$	**[Hyp]**
2.	$\Box\big((\phi_1 \wedge \phi_2) \rightarrow \phi_1\big)$	**[LTaut]**
3.	$\Box\big((\phi_1 \wedge \phi_2) \rightarrow \phi_2\big)$	**[LTaut]**
4.	$\Box\phi_1$	**[LMP(1,2)]**
5.	$\Box\phi_2$	**[LMP(1,3)]**
6.	$(\Box\phi_1) \sqcap (\Box\phi_2)$	**[GCii(4,5)]**

(ii) Prove that $(\Box\phi_1) \sqcap (\Box\phi_2) \vdash \Box(\phi_1 \wedge \phi_2)$:

1.	$(\Box\phi_1) \sqcap (\Box\phi_2)$	**[Hyp]**
2.	$\Box\phi_1$	**[GCi(1)]**
3.	$\Box\phi_2$	**[GCi(1)]**
4.	$\Box(\phi_1 \rightarrow (\phi_2 \rightarrow (\phi_1 \wedge \phi_2)))$	**[LTaut]**
5.	$\Box(\phi_2 \rightarrow (\phi_1 \wedge \phi_2))$	**[LMP(2,4)]**
6.	$\Box(\phi_1 \wedge \phi_2)$	**[LMP(3,5)]**

□

All the results so far demonstrated are necessary to prove the following theorem that says that if two local formulas are equivalent, then their probabilities must be the same.

Theorem 3.3.7. **[Eqv]** $\Box(\phi_1 \leftrightarrow \phi_2) \vdash (\int\phi_1 = \int\phi_2)$

Proof. Note that $\Box(\phi_1 \leftrightarrow \phi_2)$ is an abbreviation of $\Box(\phi_1 \rightarrow \phi_2 \wedge \phi_2 \rightarrow \phi_1)$, and that the equality $(\int\phi_1 = \int\phi_2)$ is an abbreviation of $(\int\phi_1 \leq \int\phi_2) \sqcap (\int\phi_2 \leq \int\phi_1)$. Then, we have the following derivation:

1.	$\Box(\phi_1 \rightarrow \phi_2 \wedge \phi_2 \rightarrow \phi_1)$	**[Hyp]**
2.	$\Box(\phi_1 \rightarrow \phi_2 \wedge \phi_2 \rightarrow \phi_1) \sqsupset \big(\Box(\phi_1 \rightarrow \phi_2) \sqcap \Box(\phi_2 \rightarrow \phi_1)\big)$	**[Conj]**
3.	$\Box(\phi_1 \rightarrow \phi_2) \sqcap \Box(\phi_2 \rightarrow \phi_1)$	**[GMP(1,2)]**
4.	$\Box(\phi_1 \rightarrow \phi_2)$	**[GCi(3)]**
5.	$\Box(\phi_2 \rightarrow \phi_1)$	**[GCi(3)]**
6.	$\Box(\phi_1 \rightarrow \phi_2) \sqsupset (\int\phi_1 \leq \int\phi_2)$	**[Kol3]**
7.	$\Box(\phi_2 \rightarrow \phi_1) \sqsupset (\int\phi_2 \leq \int\phi_1)$	**[Kol3]**
8.	$\int\phi_1 \leq \int\phi_2$	**[GMP(4,6)]**
9.	$\int\phi_2 \leq \int\phi_1$	**[GMP(5,7)]**

10.	$(\int\phi_1 \leq \int\phi_2) \sqcap (\int\phi_2 \leq \int\phi_1)$	**[GCii(8,9)]**

□

It is easy to check (semantically) that the opposite direction is not vaild. For example, considering $\int p = 1/2$, we have that $\int p = \int(\sim p)$, although $p \leftrightarrow (\sim p)$ is not valid in CPL.

Now our goal is to demonstrate that the probability operator of this calculus, such as it happens in Probability Theory, gives values between 0 and 1. The upper bound is the easiest to show and its proof is below.

Theorem 3.3.8. **[UB]** $\vdash \int\phi \leq 1$

Proof.

1.	$\vdash\ \Box(\phi \to \top)$	**[LTaut]**
2.	$\vdash\ \Box(\phi \to \top) \sqsupset (\int\phi \leq \int\top)$	**[Kol3]**
3.	$\vdash\ \int\phi \leq \int\top$	**[GMP(1,2)]**
4.	$\vdash\ \int\top = 1$	**[Kol1]**
5.	$\vdash\ \int\phi \leq 1$	**[Real(3,4)]**

□

The following two propositions are extremely important in this calculus and essential to prove not only the lower bound result but also a large number of other properties. Note that this part of the dissertation is only intended to show how EPPL behaves similarly to Probability Theory.

Proposition 3.3.9. **[Box]** $(\Box\phi) \vdash (\int\phi = 1)$

Proof. We have the following proof of $(\int\phi = 1)$ from $\{\Box\phi\}$:

1.	$\Box\phi$	**[Hyp]**
2.	$\Box\big(\phi \to (\top \to \phi)\big)$	**[LTaut]**
3.	$\Box(\top \to \phi)$	**[LMP(1,2)]**
4.	$\Box(\top \to \phi) \sqsupset (\int\top \leq \int\phi)$	**[Kol3]**
5.	$\int\top \leq \int\phi$	**[GMP(3,4)]**
6.	$\int\top = 1$	**[Kol1]**
7.	$1 \leq \int\phi$	**[Real(5,6)]**
8.	$\int\phi \leq 1$	**[UB]**
9.	$\int\phi = 1$	**[GCii(7,8)]**

□

Proposition 3.3.10. **[Neg]** $\vdash \int(\sim\phi) = 1 - \int\phi$

Proof.

1.	$\vdash \Box\big(\sim(\sim\phi \wedge \phi) \leftrightarrow \top\big)$	**[LTaut]**
2.	$\vdash \int\sim(\sim\phi \wedge \phi) = \int\top$	**[Eqv(1)]**
3.	$\vdash \int\top = 1$	**[Kol1]**
4.	$\vdash \int\sim(\sim\phi \wedge \phi) = 1$	**[Real(2,3)]**
5.	$\vdash \big(\int\sim(\sim\phi \wedge \phi) = 1\big) \sqsupset \big(\int(\sim\phi \vee \phi) = \int(\sim\phi) + \int\phi\big)$	**[Kol2]**
6.	$\vdash \int(\sim\phi \vee \phi) = \int(\sim\phi) + \int\phi$	**[GMP(4,5)]**
7.	$\vdash \Box(\sim\phi \vee \phi)$	**[LTaut]**
8.	$\vdash \int(\sim\phi \vee \phi) = 1$	**[Box(7)]**
9.	$\vdash 1 = \int(\sim\phi) + \int\phi$	**[Real(6,8)]**
10.	$\vdash \int(\sim\phi) = 1 - \int\phi$	**[Real(9)]**

□

An immediate consequence of this Proposition, considering that $\perp$ is an abbreviation of $(\sim\top)$, is that: **[Fal]** $\vdash \int\perp = 0$.

Given that, we can now prove the lower bound result, *i.e.* the probability of local formula ϕ is greater than or equal to zero.

Theorem 3.3.11. **[LB]** $\vdash 0 \leq \int\phi$

Proof.

1.	$\vdash \Box(\perp \rightarrow \phi)$	**[LTaut]**
2.	$\vdash \Box(\perp \rightarrow \phi) \sqsupset (\int\perp \leq \int\phi)$	**[Kol3]**
3.	$\vdash \int\perp \leq \int\phi$	**[GMP(1,2)]**
4.	$\vdash \int\perp = 0$	**[Fal]**
5.	$\vdash 0 \leq \int\phi$	**[Real(3,4)]**

□

Now joining the two previous Theorems, we have the following EPPL theorem:
[Int] $\vdash (0 \leq \int\phi) \sqcap (\int\phi \leq 1)$.

The next result can be generalizated to any finite number of formulas n. Here it is only proved to $n = 3$.

Proposition 3.3.12.
$\Box\big(\sim(\phi_1 \wedge \phi_2)\big), \Box\big(\sim(\phi_1 \wedge \phi_3)\big), \Box\big(\sim(\phi_2 \wedge \phi_3)\big) \vdash \int(\phi_1 \vee \phi_2 \vee \phi_3) = \int\phi_1 + \int\phi_2 + \int\phi_3$.

Proof.

1.	$\Box\big(\sim(\phi_1 \wedge \phi_2)\big)$	**[Hyp]**
2.	$\Box\big(\sim(\phi_1 \wedge \phi_3)\big)$	**[Hyp]**
3.	$\Box\big(\sim(\phi_2 \wedge \phi_3)\big)$	**[Hyp]**
4.	$\int\big(\sim(\phi_1 \wedge \phi_2)\big) = 1$	**[Box(1)]**
5.	$\int(\phi_1 \vee \phi_2) = \int\phi_1 + \int\phi_2$	**[Kol2+GMP(4)]**
6.	$\Box\big(\sim(\phi_1 \wedge \phi_3) \wedge \sim(\phi_2 \wedge \phi_3)\big)$	**[Conj(2,3)]**
7.	$\Box\big((\sim(\phi_1 \wedge \phi_3) \wedge \sim(\phi_2 \wedge \phi_3)) \rightarrow \sim((\phi_1 \vee \phi_2) \wedge \phi_3))\big)$	**[LTaut]**
8.	$\Box\big(\sim((\phi_1 \vee \phi_2) \wedge \phi_3))\big)$	**[LMP(7,8)]**
9.	$\int(\sim((\phi_1 \vee \phi_2) \wedge \phi_3))) = 1$	**[Box(8)]**
10.	$\int\big((\phi_1 \vee \phi_2) \vee \phi_3)\big) = \int(\phi_1 \vee \phi_2) + \int\phi_3$	**[Kol2+GMP(9)]**
11.	$\int(\phi_1 \vee \phi_2 \vee \phi_3) = \int\phi_1 + \int\phi_2 + \int\phi_3$	**[Real(5,10)]**

□

Finally, to end this section of EPPL calculus proofs, we will prove another common result of Probability Theory that is the probability of $\phi_1 \vee \phi_2$ plus the probability of $\phi_1 \wedge \phi_2$ equals to the sum of probabilities of ϕ_1 and ϕ_2.

First, we will demonstrate two lemmas that will help us on the proof of our main result.

Lemma 3.3.13. **[Aux1]** $\vdash \int(\phi_1 \vee \phi_2) = \int\phi_1 + \int(\sim\phi_1 \wedge \phi_2)$

Proof.

1.	$\vdash \Box\big((\phi_1 \vee \phi_2) \leftrightarrow (\phi_1 \vee (\sim\phi_1 \wedge \phi_2)\big)$	**[LTaut]**
2.	$\vdash \int(\phi_1 \vee \phi_2) = \int\big(\phi_1 \vee (\sim\phi_1 \wedge \phi_2)\big)$	**[Eqv(1)]**
3.	$\vdash \Box\big(\sim(\phi_1 \wedge (\sim\phi_1 \wedge \phi_2))\big)$	**[LTaut]**
4.	$\vdash \int\big(\sim(\phi_1 \wedge (\sim\phi_1 \wedge \phi_2))\big) = 1$	**[Box(3)]**
5.	$\vdash \int\big(\phi_1 \vee (\sim\phi_1 \wedge \phi_2)\big) = \int\phi_1 + \int(\sim\phi_1 \wedge \phi_2)$	**[Kol2+GMP(4)]**
6.	$\vdash \int(\phi_1 \vee \phi_2) = \int\phi_1 + \int(\sim\phi_1 \wedge \phi_2)$	**[Real(2,5)]**

□

Lemma 3.3.14. **[Aux2]** $\vdash \int\phi_2 = \int(\phi_1 \wedge \phi_2) + \int(\sim\phi_1 \wedge \phi_2)$

Proof.

1.	$\vdash \Box\big(\phi_2 \leftrightarrow ((\phi_1 \wedge \phi_2) \vee (\sim\phi_1 \wedge \phi_2))\big)$	**[LTaut]**
2.	$\vdash \int\phi_2 = \int\big((\phi_1 \wedge \phi_2) \vee (\sim\phi_1 \wedge \phi_2)\big)$	**[Eqv(1)]**
3.	$\vdash \Box\big(\sim((\phi_1 \wedge \phi_2) \wedge (\sim\phi_1 \wedge \phi_2))\big)$	**[LTaut]**
4.	$\vdash \int\big(\sim((\phi_1 \wedge \phi_2) \wedge (\sim\phi_1 \wedge \phi_2))\big) = 1$	**[Box(3)]**
5.	$\vdash \int\big((\phi_1 \wedge \phi_2) \vee (\sim\phi_1 \wedge \phi_2)\big) = \int(\phi_1 \wedge \phi_2) + \int(\sim\phi_1 \wedge \phi_2)$	**[Kol2+GMP(4)]**
6.	$\vdash \int\phi_2 = \int(\phi_1 \wedge \phi_2) + \int(\sim\phi_1 \wedge \phi_2)$	**[Real(2,5)]**

□

Therefore, by joining these two Lemmas it is possible to prove the sum theorem using only properties from real numbers (as follows).

Theorem 3.3.15. **[Sum]** $\vdash \int\phi_1 + \int\phi_2 \;=\; \int(\phi_1 \vee \phi_2) + \int(\phi_1 \wedge \phi_2)$

Proof.

1.	$\vdash \int(\phi_1 \vee \phi_2) = \int\phi_1 + \int(\sim\phi_1 \wedge \phi_2)$	**[Aux1]**
2.	$\vdash \int\phi_2 \;=\; \int(\phi_1 \wedge \phi_2) + \int(\sim\phi_1 \wedge \phi_2)$	**[Aux2]**
3.	$\vdash \int(\phi_1 \vee \phi_2) - \int\phi_1 = \int\phi_2 - \int(\phi_1 \wedge \phi_2)$	**[Real(1,2)]**
4.	$\vdash \int\phi_1 + \int\phi_2 \;=\; \int(\phi_1 \vee \phi_2) + \int(\phi_1 \wedge \phi_2)$	**[Real(3)]**

□

The next example illustrates how this Hilbert calculus can be used in factual problems. Perhaps EPPL greatest advantage is the ability of bringing into the Probability Theory the formal logical reasoning. This example was adapted from an example of probabilistic linear temporal logic in [BRS06], and here it is considered a simpler version adapted to EPPL.

Example 3.3.16 (Zero-Knowledge Protocol [MSS05])**.** In cryptography, zero-knowledge protocol is a method that allows a Prover to show to some Verifier that he knows a secret, without revealing it. There are three important properties of a zero-knowledge protocol:

(i) *Completeness*: the Verifier always accepts the proof if the Prover knows the secret;

(ii) *Soundness*: the Verifier cannot always accept the proof if the Prover does not know the secret;

(iii) *Zero-Knowledge*: Verifier cannot learn the secret.

To illustrate, we will considerer the Ali Baba cave example (simplified to EPPL). Alice (Prover) and Bob (Verifier) get in a cave that has only two paths, and those are connected inside the cave by a locked door. Alice wants to show that she knows the secret to open that door, deep inside the cave, but without telling Bob how she did it.

Initially, Alice chooses one of the paths (without Bob seeing which path she chose). Then, Bob enters the cave and asks Alice to exit for one of the paths (Bob chooses a random path). If Alice knows the secret, she can always get out for the path that he chose. If not, she has to exit from the same path that she picked initially. Bob would only not accept that she knows the secret if she came out from the path that he did not ask. This problem is illustrated in the Figure 3.1 (from wikipedia).

Let $\{p_1, p_2, p_3\}$ be the CPL propositional symbols for this problem, where:
p_1 - *Alice knows the secret*;
p_2 - *Alice gets out for the path that Bob asked*;
p_3 - *Bob accepts that she knows the secret.*

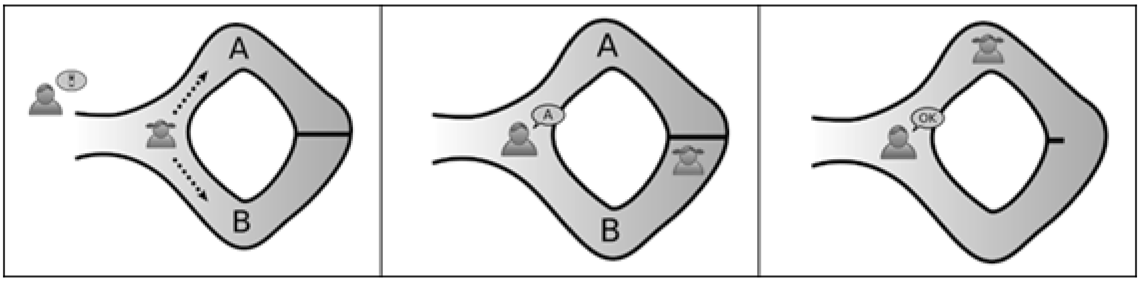

Figure 3.1: Graphical illustration of Ali Baba cave problem (wikipedia).

The specification Δ of this zero-knowledge protocol is the following:

[Hyp1] $\Box\big((p_1 \wedge p_2) \rightarrow p_3\big)$
[Hyp2] $\Box\big((p_1 \wedge \sim p_2) \rightarrow p_3\big)$
[Hyp3] $\Box\big((\sim p_1 \wedge p_2) \rightarrow p_3\big)$
[Hyp4] $\Box\big((\sim p_1 \wedge \sim p_2) \rightarrow \sim p_3\big)$
[Hyp5] $\int p_2 = 0.5$

Proof of $\Delta \cup \{\Box p_1\} \vdash \Box p_3$ (*Protocol Completeness*):

1.	$\Box p_1$	**[Hyp]**
2.	$\Box((p_1 \wedge p_2) \rightarrow p_3)$	**[Hyp1]**
3.	$\Box((p_1 \wedge \sim p_2) \rightarrow p_3)$	**[Hyp2]**
4.	$\Box(((p_1 \wedge p_2) \rightarrow p_3) \rightarrow (((p_1 \wedge \sim p_2) \rightarrow p_3) \rightarrow (p_1 \rightarrow p_3)))$	**[LTaut]**
5.	$\Box(((p_1 \wedge \sim p_2) \rightarrow p_3) \rightarrow (p_1 \rightarrow p_3))$	**[LMP(2,4)]**
6.	$\Box(p_1 \rightarrow p_3)$	**[LMP(3,5)]**
7.	$\Box p_3$	**[LMP(1,6)]**

Proof of $\Delta \cup \{\Box(\sim p_1)\} \vdash \neg(\Box p_3)$ (*Protocol Soundness*):

1.	$\Box(\sim p_1)$	**[Hyp]**
2.	$\Box((\sim p_1 \wedge p_2) \rightarrow p_3)$	**[Hyp3]**
3.	$\Box((\sim p_1 \wedge \sim p_2) \rightarrow \sim p_3)$	**[Hyp4]**
4.	$\int p_2 = 0.5$	**[Hyp5]**
5.	$\Box(((\sim p_1 \wedge p_2) \rightarrow p_3) \rightarrow (((\sim p_1 \wedge \sim p_2) \rightarrow \sim p_3) \rightarrow (\sim p_1 \rightarrow (p_3 \rightarrow p_2))))$	**[LTaut]**
6.	$\Box(((\sim p_1 \wedge \sim p_2) \rightarrow \sim p_3) \rightarrow (\sim p_1 \rightarrow (p_3 \rightarrow p_2)))$	**[LMP(2,5)]**
7.	$\Box(\sim p_1 \rightarrow (p_3 \rightarrow p_2))$	**[LMP(3,6)]**
8.	$\Box(p_3 \rightarrow p_2)$	**[LMP(1,7)]**
9.	$\Box(p_3 \rightarrow p_2) \sqsupset (\int p_3 \leq \int p_2)$	**[Kol3]**
10.	$\int p_3 \leq \int p_2$	**[GMP(8,9)]**

11.	$\int p_3 \leq 0.5$	**[Real(4,10)]**
12.	$\neg(\int p_3 = 1)$	**[Real(11)]**
13.	$\Box p_3 \sqsupset (\int p_3 = 1)$	**[Box]**
14.	$\neg(\int p_3 = 1) \sqsupset \neg(\Box p_3)$	**[GMT(13)]**
15.	$\neg(\Box p_3)$	**[GMP(12,14)]**

We verify the third property of the zero-knowledge protocol by the way that that protocol was defined (and assuming that the Verifier follows the protocol). These and other concepts and properties of zero-knowledge proof systems are approached in [GO94].

3.4 Soundness

The objective in this section is to show the soundness of the Hilbert calculus defined in Table 3.3. In the next Theorem it is proved the weakly sound version of EPPL.

Theorem 3.4.1 (Weak Soundness)**.** Let δ be an EPPL formula. If $\vdash \delta$ then $\vDash \delta$.

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an arbitrary EPPL model. We want to prove that the axioms and the inference rule defined in Table 3.3 are semantically valid, *i.e.* the arbitrary model m satisfies each one.

[LTaut] $\vdash \Box\phi$, if ϕ is tautology in CPL:
If ϕ is tautology in CPL, then $v(\phi) = 1$ for any valuation v in CPL. In particular, for each $\omega \in \Omega$, the valuation induced by the Bernoulli random variable associated to ϕ is such that $X_\phi(\omega) = v_\omega(\phi) = 1$. Then, $\Omega_\phi = \{\omega \in \Omega : X_\phi(\omega) = 1\} = \Omega$ and this is, by definition, equivalent to $m \vDash \Box\phi$.

[GTaut] $\vdash \delta$, if δ is global tautological formula:
Suppose that δ is a tautological formula, and consider ρ an assignment to real variables. Then there exists a CPL tautology ϕ and a mapping $f : Var(\mathcal{B}) \to Form(\mathscr{L})$ (in the conditions of Definition 3.3.1) such that $\delta = f(\phi)$.
Let v_m be a CPL valuation defined as: $v_m(p) = 1$ iff $(m, \rho) \vDash f(p)$. We will prove now that $v_m(\phi) = 1$ iff $(m, \rho) \vDash f(\phi)$ (by induction over CPL formulas):

- If $\phi := \sim\phi_1$ (with $\phi_1 \in Form(\mathcal{B})$):
  $v(\phi) = 1$ iff $v(\sim\phi_1) = 1$ iff $v(\phi_1) = 0$ iff $(m, \rho) \nvDash f(\phi_1)$
  iff $(m, \rho) \vDash \neg f(\phi_1)$ iff $(m, \rho) \vDash f(\sim\phi_1)$ iff $(m, \rho) \vDash f(\phi)$;
- If $\phi := \phi_1 \to \phi_2$ (with $\phi_1, \phi_2 \in Form(\mathcal{B})$):
  $v(\phi) = 1$ iff $v(\phi_1 \to \phi_2) = 1$ iff $\big(v(\phi_1) = 0$ or $v(\phi_2) = 1\big)$
  iff $\big((m, \rho) \nvDash f(\phi_1)$ or $(m, \rho) \vDash f(\phi_2)\big)$ iff $(m, \rho) \vDash f(\phi_1) \sqsupset f(\phi_2)$
  iff $(m, \rho) \vDash f(\phi_1 \to \phi_2)$ iff $(m, \rho) \vDash f(\phi)$;

Since ϕ is a local tautology, then $v_m(\phi) = 1$. Therefore, we have that $(m, \rho) \vDash f(\phi)$ and, consecutively, $(m, \rho) \vDash \delta$.

[Imp] $\vdash \Box(\phi_1 \to \phi_2) \sqsupset (\Box\phi_1 \sqsupset \Box\phi_2)$:

$m \vDash \Box(\phi_1 \to \phi_2) \sqsupset (\Box\phi_1 \sqsupset \Box\phi_2)$
iff $m \nvDash \Box(\phi_1 \to \phi_2)$ or $m \vDash (\Box\phi_1 \sqsupset \Box\phi_2)$
iff $m \nvDash \Box(\phi_1 \to \phi_2)$ or $(m \nvDash \Box\phi_1$ or $m \vDash \Box\phi_2)$
iff $m \nvDash \Box\phi_1$ or $m \nvDash \Box(\phi_1 \to \phi_2)$ or $m \vDash \Box\phi_2$
If $\big(m \nvDash \Box\phi_1$ or $m \nvDash \Box(\phi_1 \to \phi_2)\big)$, then it is done.
Otherwise, if $m \vDash \Box\phi_1$ and $m \vDash \Box(\phi_1 \to \phi_2)$:
$m \vDash \Box\phi_1$ iff $\big(X_{\phi_1}(\omega) = 1$ for each $\omega \in \Omega\big)$
$m \vDash \Box(\phi_1 \to \phi_2)$ iff $\big(X_{\phi_1 \to \phi_2}(\omega) = 1$ for each $\omega \in \Omega\big)$
Given any $\omega \in \Omega$, we have that $X_{\phi_1}(\omega) = 1$ and $X_{\phi_1 \to \phi_2}(\omega) = \max\{1 - X_{\phi_1}(\omega), X_{\phi_2}(\omega)\} = 1$. Then, $1 = \max\{1 - X_{\phi_1}(\omega), X_{\phi_2}(\omega)\} = \max\{1-1, X_{\phi_2}(\omega)\} = \max\{0, X_{\phi_2}(\omega)\} = X_{\phi_2}(\omega)$ and therefore $\Omega_{\phi_2} = \{\omega \in \Omega : X_{\phi_2}(\omega) = 1\} = \Omega$, which is equivalent to $m \vDash \Box\phi_2$.

[EqF] $\vdash (\Box\bot) \equiv \mathbf{F}$:
$m \vDash (\Box\bot) \equiv \mathbf{F}$ iff $\big(m \vDash (\Box\bot) \sqsupset \mathbf{F}\big)$ and $\big(m \vDash \mathbf{F} \sqsupset (\Box\bot)\big)$.

Both of this statements are already proved in Proposition 3.2.15 for an arbitrary formula δ. On one hand, $\big(m \vDash (\Box\bot) \sqsupset \mathbf{F}\big)$ is proved in (iii); on the other hand, $\big(m \vDash \mathbf{F} \sqsupset (\Box\bot)\big)$ is proved in (iv).

[Kol1] $\vdash \int\top = 1$:
$m \vDash (\int\top = 1)$ iff $[\int\top]_m = [1]_m$ iff $\mathbf{P}(\Omega_\top) = 1$.

Since $X_\top(\omega) = X_{\phi\to\phi}(\omega) = \max\{1 - X_\phi(\omega), X_\phi(\omega)\} = 1$ for every $\omega \in \Omega$, then $\Omega_\top = \Omega$, and we have that $\mathbf{P}(\Omega_\top) = \mathbf{P}(\Omega) = 1$.

[Kol2] $\vdash \big(\int{\sim}(\phi_1 \wedge \phi_2) = 1\big) \sqsupset \big(\int(\phi_1 \vee \phi_2) = \int\phi_1 + \int\phi_2\big)$:
$m \vDash \big(\int{\sim}(\phi_1 \wedge \phi_2) = 1\big) \sqsupset \big(\int(\phi_1 \vee \phi_2) = \int\phi_1 + \int\phi_2\big)$
iff $m \nvDash \int{\sim}(\phi_1 \wedge \phi_2) = 1$ or $m \vDash \int(\phi_1 \vee \phi_2) = \int\phi_1 + \int\phi_2$
If $m \nvDash \int{\sim}(\phi_1 \wedge \phi_2) = 1$, then the result holds.
Otherwise, if $m \vDash \int{\sim}(\phi_1 \wedge \phi_2) = 1)$, then
$m \vDash \int{\sim}(\phi_1 \wedge \phi_2) = 1$ iff $[\int{\sim}(\phi_1 \wedge \phi_2)]_m = [1]_m$
iff $[1]_m - [\int(\phi_1 \wedge \phi_2)]_m = [1]_m$ iff $[\int(\phi_1 \wedge \phi_2)]_m = 0$

Since $[\int\phi_1]_m + [\int\phi_2]_m = [\int(\phi_1 \vee \phi_2)]_m + [\int(\phi_1 \wedge \phi_2)]_m$, then $[\int(\phi_1 \vee \phi_2)]_m = [\int\phi_1]_m + [\int\phi_2]_m$ which is equivalent to $[\int(\phi_1 \vee \phi_2)]_m = [\int\phi_1 + \int\phi_2]_m$, and thus $m \vDash \int(\phi_1 \vee \phi_2) = \int\phi_1 + \int\phi_2$.

[Kol3] $\vdash \Box(\phi_1 \to \phi_2) \sqsupset (\int\phi_1 \leq \int\phi_2)$:
$m \vDash \Box(\phi_1 \to \phi_2) \sqsupset (\int\phi_1 \leq \int\phi_2)$
iff $m \nvDash \Box(\phi_1 \to \phi_2)$ or $m \vDash \int\phi_1 \leq \int\phi_2$
iff is false that $\big(X_{\phi_1 \to \phi_2}(\omega) = 1$ for each $\omega \in \Omega\big)$ or $\big([\int\phi_1]_m \leq [\int\phi_2]_m\big)$
iff $\big(X_{\phi_1 \to \phi_2}(\omega) = 0$ for some $\omega \in \Omega\big)$ or $\big(\mathbf{P}(\Omega_{\phi_1}) \leq \mathbf{P}(\Omega_{\phi_2})\big)$
iff $\big(\max\{1 - X_{\phi_1}(\omega), X_{\phi_2}(\omega)\} = 0$ for some $\omega \in \Omega\big)$ or $\big(\mathbf{P}(\Omega_{\phi_1}) \leq \mathbf{P}(\Omega_{\phi_2})\big)$
If $\mathbf{P}(\Omega_{\phi_1}) \leq \mathbf{P}(\Omega_{\phi_2})$, then the result follows.
Otherwise, if $\mathbf{P}(\Omega_{\phi_1}) > \mathbf{P}(\Omega_{\phi_2})$, then there exists $\omega_0 \in \Omega$ such that $\omega_0 \in \Omega_{\phi_1}$ and $\omega_0 \notin \Omega_{\phi_2}$. Thus, $\max\{1 - X_{\phi_1}(\omega_0), X_{\phi_2}(\omega_0)\} = \max\{1 - 1, 0\} = 0$.

[Real] $\vdash (t_1 \leq t_2)$, for each valid analytical real inequality:
$m \vDash (t_1 \leq t_2)$ iff $[t_1]_m \leq [t_2]_m$, which is immediately true because it is a valid inequality.

[GMP] $\delta_1, (\delta_1 \sqsupset \delta_2) \vdash \delta_2$:

Suppose that $m \vDash \delta_1$ and $m \vDash \delta_1 \sqsupset \delta_2$. The latter is equivalent to ($m \nvDash \delta_1$ or $m \vDash \delta_2$) and, together with the first assumption, follow immediately that $m \vDash \delta_2$.

□

Furthermore, we can easily show that EPPL is in fact strongly Sound using the Weak Soundness Theorem.

Theorem 3.4.2 (Strong Soundness)**.** Let δ be an EPPL formula and Δ a set (possibly infinite) of global formulas. If $\Delta \vdash \delta$ then $\Delta \vDash \delta$.

Proof. Suppose that $\Delta \vdash \delta$. By definition of axiomatizable logic system (see Definition 2.2.1), there is a finite set $\Delta_0 = \{\delta_1, ..., \delta_n\} \subseteq \Delta$ such that $\Delta_0 \vdash \delta$. Then, by the Deduction Theorem (Theorem 3.3.3), it is possible to deduce that $\vdash \delta_1 \sqsupset (\delta_2 \sqsupset (... \sqsupset (\delta_n \sqsupset \delta)))$. Since EPPL is weakly Sound (by Theorem 3.4.1), it follows that $\vDash \delta_1 \sqsupset (\delta_2 \sqsupset (... \sqsupset (\delta_n \sqsupset \delta)))$. Therefore, by definiton of the EPPL semantical relation $\vDash$, we conclude that $\{\delta_1, ..., \delta_n\} \vDash \delta$, and this implies that $\Delta \vDash \delta$.

□

3.5 Completeness

This section is intended to discuss the Completeness of EPPL defined in the previous sections of this chapter. In fact, although it is possible to prove weak Completeness, unfortunately it is not possible to prove the strong Completeness version (see Definition 2.2.4) because EPPL is not a compact logic. The example below illustrates this with a counterexample.

Example 3.5.1. (This example is in [MSS05] with a similar syntax and semantics for a probabilistic logic, and here it is adapted to our logic.)

On one hand, it is possible to show that $\{(r \leq x_k) : r < 0.5\} \vDash (0.5 \leq x_k)$, but since there not exists a finite subset $S \subset \{(r \leq x_k) : r < 0.5\}$ such that $S \vDash (0.5 \leq x_k)$, then EPPL is not a compact logic. Therefore, it is impossible to prove that $\{(r \leq x_k) : r < 0.5\} \vdash (0.5 \leq x_k)$ because the rules in Hilbert calculus are finitary, and then EPPL is not strongly complete.

However, although it is very difficult to show, it is possible to prove that EPPL is weakly complete. The main goal of this section is to prove that, but first we need to give some attention to some results presented and demonstrated in [BMN10].

Lemma 3.5.2 (Small Model Theorem)**.** If δ is a satisfiable EPPL formula, then it has a finite model using at most $2|\delta| + 1$ algebraic real numbers, where $|\delta|$ represents the number of symbols required to write the formula.

Proof. See Theorem 2.8 in [BMN10]. □

We now mention several sets related to a global formula δ, in order to formulate a satisfiability algorithm to δ, that is, an algorithm that returns an EPPL model that satisfies δ (see Definition 2.2.3)

- $prop(\delta)$ - set of all CPL propositional symbols that occur in δ;
- $var(\delta)$ - set of all real logical variables that occur in δ;
- $iq(\delta)$ - set of all inequalities $(t_1 \leq t_2)$ that occur in δ;
- $bf(\delta)$ - set of all universal subformulas $\Box\phi$ that occur in δ;
- $at(\delta) = iq(\delta) \cup bf(\delta)$ - set of all global atoms of δ;
- $\varepsilon^\delta = \delta_1 \sqcap ... \sqcap \delta_k$ - exhaustive conjunction of literals of $at(\delta)$, where δ_i is either a global atom or its negation ($i = 1, ..., k$, where $k = |at(\delta)|$);
- ϕ^δ - CPL formula obtained by replacing in δ, each global atom δ_i with a fresh propositional symbol p_i (for $i = 1, ..., k$) and the global connectives $(\neg, \sqsupset)$ by the local connectives $(\sim, \rightarrow)$;
- v_{ε^δ} - CPL valuation over propositional symbols $p_1, ..., p_k$, such that $v_{\varepsilon^\delta}(p_i) = 1$ if and only if δ_i occurs positively in ε^δ;
- $lbf(\varepsilon^\delta)$ - set of CPL formulas ϕ such that $(\Box\phi)$ occurs positively in ε^δ;
- $nlbf(\varepsilon^\delta)$ - set of CPL formulas ϕ such that $(\Box\phi)$ occurs negatively in ε^δ;
- δ_0^a - denote the analytical formula where all terms $(\int\phi)$ are replaced in subformula $\delta_0 \in iq(\varepsilon^\delta)$, by
$$\sum_{v \in V : v(\phi)=1} x_v$$
where x_v is a fresh real variable.

As defined in Chapter 2, a global formula δ is satisfiable if there exists an EPPL model m and a real assignment ρ such that $(m, \rho) \vDash \delta$. The purpose of this algorithm is to find a model that satisfies δ. Obviously, because of this, the input of this algorithm is a formula δ, and the output will be an EPPL model and a real assignment if δ is satisfiable, and *no model* otherwise.

The idea behind this algorithm is trying to solve this EPPL problem using the base logic, *i.e.* make the global formulas being interpreted, in a certain way, as CPL formulas. First, we will separate the atoms of δ into two distinct sets: $iq(\delta)$ (inequalities that occur in δ) and $bf(\delta)$ (universal subformulas that occur in δ).

Then, we considered all exhaustive conjunction of literals $\varepsilon^\delta = \delta_1 \sqcap ... \sqcap \delta_k$ of $at(\delta)$ such that $v_{\varepsilon^\delta} \vDash_{\mathcal{B}} \phi^\delta$, where ϕ^δ is a CPL formula obtained by replacing in δ, each global atom δ_i with a fresh propositional symbol p_i (for $i = 1, ..., k$) and the global connectives by the corresponding local connectives; and v_{ε^δ} is a valuation over this propositional symbols, such that $v_{\varepsilon^\delta}(p_i) = 1$ if and only if δ_i occurs positively in ε^δ.

For each ε^δ under these conditions, we will determine $lbf(\varepsilon^\delta)$ (set of CPL formulas ϕ such that $(\Box\phi)$ occurs positively in ε^δ) and $nlbf(\varepsilon^\delta)$ (CPL formulas which occur negatively).

In this step of the algorithm, we will consider all sets of valuations $V \subseteq 2^{prop(\delta)}$, in the conditions of the Small Model Theorem, such that for each valuation $v \in V$, we have that $v \vDash_{\mathcal{B}} \phi_1$ for all $\phi_1 \in lbf(\varepsilon^\delta)$, and $v \nvDash_{\mathcal{B}} \phi_2$ for all $\phi_2 \in nlbf(\varepsilon^\delta)$.

Algorithm SatEPPL(δ)

In EPPL formula δ

Out $(V, \mathbf{P})$ (denoting EPPL model $m = (V, \mathcal{P}(V), \mathbf{P}, \mathbf{X})$) and assignment ρ or *no model*

compute $bf(\delta)$, $iq(\delta)$ and $at(\delta)$
for each exhaustive conjunction ε^δ of literals of $at(\delta)$ s.t. $v_{\varepsilon^\delta} \vDash_{\mathcal{B}} \phi^\delta$, **do**:
  compute $lbf(\varepsilon^\delta)$ and $nlbf(\varepsilon^\delta)$
  for each $V \subseteq 2^{prop(\delta)}$ s.t. $0 < |V| \leq 2|\delta| + 1$, $V \vDash_{\mathcal{B}} \wedge lbf(\varepsilon^\delta)$ and $V \nvDash_{\mathcal{B}} \phi$ for all $\phi \in nlbf(\varepsilon^\delta)$, **do**:
    $\kappa \longleftarrow \big(\sum_{v \in V} x_v = 1\big) \sqcap \big(\sqcap_{v \in V} (0 \leq x_v)\big)$
    for each $\delta_0 \in iq(\delta)$, **do**:
      $\kappa \longleftarrow \kappa \sqcap \delta_0^a$
    end
    $\mathcal{R} = SatReal(\kappa)$
    if $\mathcal{R}$ is *real model*,
      $\mathbf{P} \longleftarrow \mathcal{R}|_{\{x_v : v \in V\}}$
      $\rho \longleftarrow \mathcal{R}|_{var(\delta)}$
      return $(V, \mathbf{P})$ and ρ
    end
  end
end
return *no model*

Table 3.3: EPPL Satisfaction Algorithm

For any set of CPL valuations V (under the previous conditions), we consider a real formula $\kappa := \big(\sum_{v \in V} x_v = 1\big) \sqcap \big(\sqcap_{v \in V} (0 \leq x_v)\big)$. These conditions characterize a probability space (the sum of all probabilities of sample points is one, and each probability is not negative). Note that V will be returned as the sample space of the EPPL model.

Moreover, for each $\delta_0 \in iq(\delta)$, we will add the analytical formula δ_0^a where all terms $(\int \phi)$ are replaced by the sum of all x_v (with $v \in V$) such that $v \vDash_{\mathcal{B}} \phi$.

We now make use of a satisfaction algorithm for real numbers (*SatReal*). If this algorithm returns a real model for κ, then we return the EPPL model $m = \big(V, \mathcal{P}(V), \mathbf{P}, \mathbf{X}\big)$ and the real assignment ρ given by $SatReal(\kappa)$. Otherwise, we will try other set of CPL valuations V in the conditions referred above. In case the algorithm runs until the end without returning any EPPL model, we conclude that δ is not satisfiable and the algorithm give us *no model*.

The following examples show us how this algorithm works.

Example 3.5.3. Consider the EPPL formula $\delta := \Box(p \vee q) \sqsupset \big((\int p \leq 0.5) \sqcap (x \leq 1)\big)$.

The first step of the algorithm gives us the following sets of subformulas of δ:

- $bf(\delta) = \{\Box(p \vee q)\}$;
- $iq(\delta) = \{(\int p \leq 0.5), (x \leq 1)\}$;
- $at(\delta) = bf(\delta) \cup iq(\delta) = \{\Box(p \vee q), (\int p \leq 0.5), (x \leq 1)\}$.

Moreover $prop(\delta) = \{p, q\}$, and an exhaustive conjunction of literals over $at(\delta)$ is $\varepsilon^\delta = \gamma_1 \sqcap \gamma_2 \sqcap \gamma_3$, where γ_1 is $\delta_1 = \Box(p \vee q)$ (or its negation), γ_2 is $\delta_2 = (\int p \leq 0.5)$ (or its negation) and γ_3 is $\delta_3 = (x \leq 1)$ (or its negation).

We have to analyze when is the valuation v_{ε^δ} of the CPL formula $\phi^\delta = p_1 \to (p_2 \wedge p_3)$ such that $v_{\varepsilon^\delta} \vDash_{\mathcal{B}} \phi^\delta$. There are five cases that satisfy it. Those are:

- $\varepsilon_1 = \delta_1 \sqcap \delta_2 \sqcap \delta_3$;
- $\varepsilon_2 = (\neg\delta_1) \sqcap \delta_2 \sqcap \delta_3$;
- $\varepsilon_3 = (\neg\delta_1) \sqcap \delta_2 \sqcap (\neg\delta_3)$;
- $\varepsilon_4 = (\neg\delta_1) \sqcap (\neg\delta_2) \sqcap \delta_3$;
- $\varepsilon_5 = (\neg\delta_1) \sqcap (\neg\delta_2) \sqcap (\neg\delta_3)$.

For $\varepsilon_1 = \delta_1 \sqcap \delta_2 \sqcap \delta_3$, we have that $lbf(\varepsilon_1) = \{(p \vee q)\}$ and $nlbf(\varepsilon_1) = \emptyset$. In the next step of the algorithm, we have to find a CPL set of valuations $V \subseteq 2^{\{p,q\}}$ such that $V \vDash_{\mathcal{B}} (p \vee q)$. There exists seven sets (we will denote each valuation as a pair where $(v(p), v(q))$): $V_1 = \{(1,1)\}$, $V_2 = \{(1,0)\}$, $V_3 = \{(0,1)\}$, $V_4 = \{(1,1),(1,0)\}$, $V_5 = \{(1,1),(0,1)\}$, $V_6 = \{(1,0),(0,1)\}$ and $V_7 = \{(1,1),(1,0),(0,1)\}$. The algorithm will now test each set.

For example, to $V_5 = \{(1,1),(0,1)\}$ (x_1 and x_2 represent $x_{(1,1)}$ and $x_{(0,1)}$, respectively), the algorithm first assigns: $\kappa = (x_1 + x_2 = 1) \sqcap (0 \leq x_1) \sqcap (0 \leq x_2)$.

Now, replacing $(\int p)$ by $\sum_{v \in V_5 : v(p)=1} x_v = x_1$, the algorithm will assign to κ the inequality atoms: $\kappa = (x_1 + x_2 = 1) \sqcap (0 \leq x_1) \sqcap (0 \leq x_2) \sqcap (x_1 \leq 0.5) \sqcap (x \leq 1)$.

In this step, this algorithm makes use of an algorithm of Satisfaction in reals that gives us a real solution that satisfies κ (or *no solution*). For this κ, we have that, for example, $x = 1$, $x_1 = x_2 = 0.5$ is a solution.

Then, an EPPL model that satisfies δ is $m = (\Omega, \mathcal{P}(\Omega), \mathbf{P}, \mathbf{X})$, where $\Omega = \{(1,1),(0,1)\}$, $\mathbf{P}(\{(1,1)\}) = \mathbf{P}(\{(0,1)\}) = 0.5$, and ρ is an assignment such that $\rho(x) = 1$.

In fact, it is easy to show that this model satisfies semantically the EPPL formula δ:

$m \vDash \Box(p \vee q) \sqsupset ((\int p \leq 0.5) \sqcap (x \leq 1))$
iff $m \nvDash \Box(p \vee q)$ or $m \vDash ((\int p \leq 0.5) \sqcap (x \leq 1))$
iff $m \nvDash \Box(p \vee q)$ or $(m \vDash (\int p \leq 0.5)$ and $m \vDash (x \leq 1))$
iff $m \nvDash \Box(p \vee q)$ or $([\int p]_m \leq [0.5]_m$ and $[x]_m \leq [1]_m)$
iff $m \nvDash \Box(p \vee q)$ or $([\int p]_m \leq [0.5]_m$ and $[x]_m \leq [1]_m)$
iff $m \nvDash \Box(p \vee q)$ or $(\mathbf{P}(\{\omega \in \Omega : X_p(\omega) = 1\}) \leq 0.5$ and $\rho(x) \leq 1)$
iff $m \nvDash \Box(p \vee q)$ or $(\mathbf{P}(\{(1,1)\}) \leq 0.5$ and $\rho(x) \leq 1)$
iff $m \nvDash \Box(p \vee q)$ or $(0.5 \leq 0.5$ and $1 \leq 1)$, which is a logically true statement.

Example 3.5.4. Considerer now the EPPL formula $\delta := (\Box p) \sqcap (\int p \leq 0.5)$.

The first step of the algorithm gives us the following sets of subformulas in δ:

- $bf(\delta) = \{\Box p\}$;
- $iq(\delta) = \{\int p \leq 0.5\}$;
- $at(\delta) = bf(\delta) \cup iq(\delta) = \{\Box p, (\int p \leq 0.5)\}$.

Moreover, $prop(\delta) = \{p\}$, and an exhaustive conjunction of literals over $at(\delta)$ is $\varepsilon^\delta = \gamma_1 \sqcap \gamma_2$, where γ_1 is $\delta_1 = \Box p$ (or its negation) and γ_2 is $\delta_2 = (\int p \leq 0.5)$ (or its negation). We have to analyze when is the valuation v_{ε^δ} of the respective local formula $\phi^\delta = (p_1 \wedge p_2)$ such that $v_{\varepsilon^\delta} \vDash_{\mathcal{B}} \phi^\delta$. There is only one conjunction of literals that makes this valid: $\varepsilon_1 = \delta_1 \sqcap \delta_2$.

For this ε_1, we have that $lbf(\varepsilon_1) = \{p\}$ and $nlbf(\varepsilon_1) = \emptyset$. In the next step of the algorithm, we have to find a CPL set of valuations $V \subseteq 2^{\{p\}}$ such that $V \vDash_{\mathcal{B}} p$. In this case, $V = \{v_1\}$, where v_1 is a valuation such that $v_1(p) = 1$.

The algorithm first assigns: $\kappa := (x_{v_1} = 1) \sqcap (0 \leq x_{v_1})$.

Now, replacing $(\int p)$ by $\sum_{v \in V: v(p)=1} x_v = x_{v_1}$, the algorithm will assign to κ the inequality atoms, that is: $\kappa := (x_{v_1} = 1) \sqcap (0 \leq x_{v_1}) \sqcap (x_{v_1} \leq 0.5)$.

In this step, this algorithm makes use of an algorithm of Satisfaction in reals that gives us a real solution that satisfies κ (or gives *no solution*). For this κ, will give us *no solution.*

Since ε^{δ} is the only exhaustive conjunction of literals that we have to consider, the algorithm returns *no model.*

In the first example δ is a satisfiable EPPL formula, and in the second δ is not satisfiable. The following Lemma says that the algorithm is Table 3.3 give us an EPPL model that satisfies a formula δ if and only if there exists such EPPL model.

Lemma 3.5.5. The algorithm in Table 3.3 decides the satisfiability of an EPPL formula.

Proof. See Theorem 2.9 in [BMN10]. □

The main purpose of including this algorithm in this work is to show that EPPL is weakly complete. Since it is not possible to prove the strong version of completeness, this result has a vital importance for EPPL.

Theorem 3.5.6 (Weak Completeness)**.** The Hilbert calculus defined in Table 3.2 is a weakly complete axiomatization of EPPL, that is, if $\vDash \delta$ then $\vdash \delta$.

Proof. Our goal is prove that, given δ a global formula, if $\vDash \delta$ then $\vdash \delta$. In this proof, it will be used a contrapositive argument, *i.e.* suppose that $\nvdash \delta$. We want to prove that $\nvDash \delta$:

A formula δ_0 is called consistent if $\nvdash \neg\delta_0$. Observe that if $\nvdash \delta$ then $\nvdash \neg(\neg\delta)$, that is, $\neg\delta$ is consistent. If $\neg\delta$ is consistent and it has a model, then $\nvDash \delta$. Therefore, we need to prove that every consistent formula has a model.

Suppose that δ is consistent and, by contradiction, the EPPL Satisfaction algorithm returns *no model.*

Let $E = \{\varepsilon : \varepsilon$ is an exhaustive conjunction of literals such that $v_{\varepsilon^\delta} \vDash_{\mathcal{B}} \phi^{\delta}\}$.

The CPL formula $\phi := \vee_{\varepsilon \in E}(\phi^{\varepsilon}) \leftrightarrow \phi^{\delta}$ is a tautology and, by the completeness of CPL, we have that $\vdash_{\mathcal{B}} \vee_{\varepsilon \in E}(\phi^{\varepsilon}) \leftrightarrow \phi^{\delta}$. And by **[GTaut]**, we have that $\vdash \sqcup_{\varepsilon \in E}(\varepsilon) \equiv \delta$. If δ is consistent then there is an exhaustive conjunction of literals ε which is also consistent, and if δ has no EPPL model, then ε has no EPPL model as well. If when running this consistent ε in the satisfaction algorithm (in Table 3.5), it returns *no model*, then it has to be for one of the following two reasons:

(a) it can not find V (line 4);

(b) for all viable V, the SatReal algorithm returns *no model* (line 9).

In both cases, it is possible (see below) to contradict the consistency of δ and then it is proved the weak completeness of EPPL.

Case (a): even if we remove the upper bound to $|V|$ (given by Small Model Theorem), the algorithm has to fail. We consider all possible sets of CPL valuations. In particular, if we take $V = 2^{prop(\delta)}$, we have that $V \nvDash_{\mathcal{B}} \wedge lbf(\varepsilon)$ or $V \vDash_{\mathcal{B}} \phi$ for some $\phi \in nlbf(\varepsilon)$. If $V \nvDash_{\mathcal{B}} \wedge lbf(\varepsilon)$, then $V \nvDash_{\mathcal{B}} \phi$ for some $\phi \in lbf(\varepsilon)$, which is equivalent to $\vDash_{\mathcal{B}} (\phi \to \perp)$. By **[LTaut]**, we have that $\vdash \Box(\phi \to \perp)$. With the axioms **[Imp]** and **[EqF]** and the inference rules **[GMP]** and **[GMT]**, we can derivate that $\vdash \neg(\Box\phi)$, and then $\nvdash \neg\varepsilon$ (contradiction!). And if $V \vDash_{\mathcal{B}} \phi$

for some $\phi \in nlbf(\varepsilon)$, then $\vdash \Box\phi$ by **[LTaut]** and considering that ϕ is a formula such that $\neg(\Box\phi)$ occurs in ε, we have that $\nvdash \neg\varepsilon$ (contradiction!).

Case (b): we need to prove that the algorithm will also fail for all viable sets V, that is, for all sets of CPL valuations V such that $V \vDash_{\mathcal{B}} \wedge lbf(\varepsilon)$ or $V \nvDash_{\mathcal{B}} \phi$ for all $\phi \in nlbf(\varepsilon)$. We can see that the sets of valuations in this conditions are closed under unions, and then there exists a set $V_{\max}$ under this condition that contains all the others, and for this set $V_{\max}$ the algorithm would fail at line 9. Let $V^c = 2^{prop(\delta)} \setminus V_{\max}$. Since ε is consistent, then the formula $\varepsilon_0 := \varepsilon \sqcap \big(\sqcap_{v \in V^c} \Box(\sim \phi_v)\big)$ is also consistent, where ϕ_v is a CPL formula that is satisfied only by the valuation $v \in V^c$. Therefore, $\vdash \varepsilon_0 \sqsupset \varepsilon$. Moreover, for each $v \in V^c$, $\vdash_{\mathcal{B}} \wedge lbf(\varepsilon) \rightarrow (\sim \phi_v)$, and we can derive that $\vdash \big(\sqcap_{\phi \in lbf(\varepsilon)} \Box\phi\big) \sqsupset \big(\sqcap_{v \in V^c} \Box(\sim \phi_v)\big)$, and so $\vdash \varepsilon \sqsupset \varepsilon_0$, from which we conclude that $\vdash \varepsilon_0 \equiv \varepsilon$. Thus, since ε is also consistent, then ε_0 is consistent, and if there is no model for ε then there is no model for ε_0 as well, and the algorithm must fail (*i.e.* return *no model*) in the line where it returns a model (that is, line 9). It is possible to prove that ε_0 is not consistent (see details in [BMN10]), and then ε is not consistent (contradiction!).

□

Chapter 4

Extending Probabilistic Propositional Logic

Considering the entire Chapter 3, we now assume all the results previously showed. Namely, the theorems of Soundness and weakly Completeness of EPPL.

In this chapter, we will discuss some more concepts and properties of Probability Theory and of other probabilistic logics that, in the context of EPPL, as far as we know have not yet been considered. Namely, we will adapt the concepts of *uncertainty* (see [Ada98]) and *interval probability* (see [Hai96]) to EPPL, and prove some features about it. Next we introduce the concept of conditional in EPPL, and discusse two ways to do this: the first consists of introducing the conditional at a higher level (keeping unchanged the base logic); and the other consists of putting conditional formulas in the base logic (called *Supposicional logic*). Finally, we will present a generalization of the probabilization process of propositional logic, but now for any satisfaction logic system that we want to make probabilistic.

4.1 Uncertainty

The *uncertainty* of a local formula will represent the probability of the formula being false, which is 1 minus its probability. The uncertainty of ϕ will be written as $\fint\phi := 1 - \int\phi$. It is not quite a new concept, instead it is more an abbreviation.

Some results with uncertainty follow directly from the ones proved in Chapter 3. The statements below illustrate some of these results (the proof is immediate considering the similar previous results).

[EqvU] $\Box(\phi_1 \leftrightarrow \phi_2) \vdash (\fint\phi_1 = \fint\phi_2)$

[BoxU] $(\Box\phi) \vdash (\fint\phi = 0)$

[IntU] $\vdash (0 \leq \fint\phi) \sqcap (\fint\phi \leq 1)$

[SumU] $\vdash \fint\phi_1 + \fint\phi_2 = \fint(\phi_1 \vee \phi_2) + \fint(\phi_1 \wedge \phi_2)$

Considering that EPPL is a sound logic, the following results will be proved using the Hilbert calculus for EPPL, but they are also valid in a semantical context (for every EPPL model).

This concept of uncertainty and the following results are discussed in [Ada98], but within a context not so formal as EPPL and from a slightly more philosophical point of view.

The following theorem says that the uncertainty of a conjunction of CPL formulas is, at most, the sum of their uncertainties.

Theorem 4.1.1. **[AndU]** $\vdash \fint(\phi_1 \wedge ... \wedge \phi_n) \leq \fint\phi_1 + ... + \fint\phi_n$, for $n \geq 2$.

Proof. In this proof we will use induction over natural numbers.
Base step ($n = 2$):

1.	$\vdash$	$\fint\phi_1 + \fint\phi_2 \;=\; \fint(\phi_1 \vee \phi_2) + \fint(\phi_1 \wedge \phi_2)$	**[SumU]**
2.	$\vdash$	$\big(0 \leq \fint(\phi_1 \vee \phi_2)\big) \sqcap \big(\fint(\phi_1 \vee \phi_2) \leq 1\big)$	**[IntU]**
3.	$\vdash$	$0 \leq \fint(\phi_1 \vee \phi_2)$	**[GCi(2)]**
4.	$\vdash$	$\fint(\phi_1 \wedge \phi_2) \leq \fint\phi_1 + \fint\phi_2$	**[Real(1,3)]**

Inductive step ($n = k$): suppose that $\vdash \fint(\phi_1 \wedge ... \wedge \phi_{k-1}) \leq \fint\phi_1 + ... + \fint\phi_{k-1}$. Then:

1.	$\vdash$	$\fint\big((\phi_1 \wedge ... \wedge \phi_{k-1}) \wedge \phi_k\big) \leq \fint(\phi_1 \wedge ... \wedge \phi_{k-1}) + \fint\phi_k$	**[BaseStep]**
2.	$\vdash$	$\fint(\phi_1 \wedge ... \wedge \phi_{k-1}) \leq \fint\phi_1 + ... + \fint\phi_{k-1}$	**[IndHyp]**
3.	$\vdash$	$\fint(\phi_1 \wedge ... \wedge \phi_{k-1} \wedge \phi_k) \leq \fint\phi_1 + ... + \fint\phi_{k-1} + \fint\phi_k$	**[Real(1,2)]**

□

Contrary to the previous result, the next proposition is a curiosity which says that if the CPL formulas $\{\phi_1, ..., \phi_n\}$ are incompatible, then the sum of their uncertainties cannot be less than one.

Proposition 4.1.2. **[IncU]** $\Box\big((\phi_1 \wedge ... \wedge \phi_n) \leftrightarrow \bot\big) \vdash \fint\phi_1 + ... + \fint\phi_n \geq 1$, for $n \geq 2$.

Proof.

1.	$\Box\big((\phi_1 \wedge ... \wedge \phi_n) \leftrightarrow \bot\big)$	**[Hyp]**
2.	$\fint(\phi_1 \wedge ... \wedge \phi_n) = \fint\bot$	**[EqvU]**
3.	$\int\bot = 0$	**[Fal]**
4.	$\fint\bot = 1$	**[Real(3)]**
5.	$\fint(\phi_1 \wedge ... \wedge \phi_n) = 1$	**[Real(2,4)]**
6.	$\fint(\phi_1 \wedge ... \wedge \phi_n) \leq \fint\phi_1 + ... + \fint\phi_n$	**[AndU]**
7.	$\fint\phi_1 + ... + \fint\phi_n \geq 1$	**[Real(5,6)]**

□

The next Theorem relates the uncertainty of the conclusion to the uncertainty of the premises of a valid inference in CPL. This gives an upper bound to the uncertainty of the conclusion that only depends of the uncertainties $\fint\phi_i$. We will see further that this upper bound can be improved to a better one.

Theorem 4.1.3 (Uncertainty Sum Theorem)**.** Let $\phi, \phi_1, ..., \phi_n$ be CPL formulas such that $\{\phi_1, ..., \phi_n\} \vDash_{\mathcal{B}} \phi$. Then $\vdash ⨍\phi \leq ⨍\phi_1 + ... + ⨍\phi_n$.

Proof. Suppose that $\phi, \phi_1, ..., \phi_n \in \text{Form}(\mathcal{B})$ such that $\{\phi_1, ..., \phi_n\} \vDash_{\mathcal{B}} \phi$. This implies that $(\phi_1 \wedge ... \wedge \phi_n) \vDash_{\mathcal{B}} \phi$, and then $\vDash_{\mathcal{B}} (\phi_1 \wedge ... \wedge \phi_n) \to \phi$. Thus, $\vdash \Box\big((\phi_1 \wedge ... \wedge \phi_n) \to \phi\big)$.

1.	$\vdash \Box\big((\phi_1 \wedge ... \wedge \phi_n) \to \phi\big)$	**[Hyp]**
2.	$\vdash \Box\big((\phi_1 \wedge ... \wedge \phi_n) \to \phi\big) \sqsupset \big(\int(\phi_1 \wedge ... \wedge \phi_n) \leq \int\phi\big)$	**[Kol3]**
3.	$\vdash \int(\phi_1 \wedge ... \wedge \phi_n) \leq \int\phi$	**[GMP(1,2)]**
4.	$\vdash ⨍\phi \leq ⨍(\phi_1 \wedge ... \wedge \phi_n)$	**[Real(3)]**
5.	$\vdash ⨍(\phi_1 \wedge ... \wedge \phi_n) \leq ⨍\phi_1 + ... + ⨍\phi_n$	**[AndU]**
6.	$\vdash ⨍\phi \leq ⨍\phi_1 + ... + ⨍\phi_n$	**[Real(4,5)]**

Since EPPL is weakly sound (Theorem 3.4.1), it follows immediatly that for every EPPL model m, we also have that $m \vDash ⨍\phi \leq ⨍\phi_1 + ... + ⨍\phi_n$.

□

Example 4.1.4. Consider the following CPL inferences:

- $\{p_1, p_1 \to p_2\} \vDash_{\mathcal{B}} p_2$:

  Suppose that m is an EPPL model such that $[⨍p_1]_m = [⨍(p_1 \to p_2)]_m = 0.1$. Then, by Theorem 4.1.3, we have that $[⨍p_2]_m \leq 0.2$.

- $\{p_1, p_2, p_3, p_4\} \vDash_{\mathcal{B}} \; p_1 \wedge (p_2 \vee p_3)$:

  Suppose in this case that m is an EPPL model such that $[⨍p_i]_m = 0.1$ (with $i = 1, 2, 3, 4$). Then, the Uncertainty Sum Theorem, for this CPL inference, give to the uncertainty of the conclusion the upper bound: $[⨍\big(p_1 \wedge (p_2 \vee p_3)\big)]_m \leq 0.4$.

In the first case, both premises are required for the inference to be a valid CPL inference. However, in the second case, there is some redundancy in the premises (for example, p_4 does not appear at all in the conclusion, and it makes no sense that its uncertainty interferes with the uncertainty of the conclusion).

The upper bound given by Theorem 4.1.3 of a CPL inference can be more refined, *i.e.* usually there will be a gap between $⨍(\phi_1 \wedge ... \wedge \phi_n)$ and $(⨍\phi_1 + ... + ⨍\phi_n)$, and then $(⨍\phi_1 + ... + ⨍\phi_n)$ is not the best upper bound to $⨍\phi$. But, usually $⨍(\phi_1 \wedge ... \wedge \phi_n)$ is unknown and we have to try to find a better bound that only considers the uncertainties of the premises ϕ_i. The next definition and the following theorem give an idea of how this could be done.

Definition 4.1.5. Given a valid CPL semantical consequence $\{\phi_1, ..., \phi_n\} \vDash_{\mathcal{B}} \phi$, the **degree of essentialness** of each premise ϕ_i is $e(\phi_i) = 1/k_i$, where k_i is the size of the smallest essential set of premises (set whose omission would make the inference invalid) to which ϕ_i belongs; and $e(\phi_i) = 0$ if ϕ_i does not belong to any essential set of premises.

Example 4.1.6. The following sentences examplifies the concept in the previous definition.

- $\{p_1, p_1 \to p_2\} \vDash_{\mathcal{B}} p_2 :\ \ e(p_1) = e(p_1 \to p_2) = 1;$
- $\{p_1, p_2\} \vDash_{\mathcal{B}} (p_1 \wedge p_2) :\ \ e(p_1) = e(p_2) = 1;$
- $\{p_1, p_2\} \vDash_{\mathcal{B}} (p_1 \vee p_2) :\ \ e(p_1) = e(p_2) = 1/2;$
- $\{p_1, p_1 \to p_2, p_1 \wedge p_2\} \vDash_{\mathcal{B}} p_2 :\ \ e(p_1) = e(p_1 \to p_2) = e(p_1 \wedge p_2) = 1/2;$
- $\{p_1, p_1 \to p_2, p_1 \wedge p_2\} \vDash_{\mathcal{B}} (p_1 \vee p_2) :\ \ e(p_1) = e(p_1 \wedge p_2) = 1/2,\ e(p_1 \to p_2) = 1/3.$
- $\{p_1, p_2, p_3, p_4\} \vDash_{\mathcal{B}}\ p_1 \wedge (p_2 \vee p_3) :\ \ e(p_1) = 1,\ e(p_2) = e(p_3) = 1/2,\ e(p_4) = 0;$

The following theorem improves the upper bound of Theorem 4.1.3 by removing some redundancy from the premises.

Theorem 4.1.7 (Essentialness Sum Theorem)**.** If $\{\phi_1, ..., \phi_n\} \vDash_{\mathcal{B}} \phi$ is a valid CPL semantical consequence, then $\vdash ⨍\phi \leq e(\phi_1) \times ⨍\phi_1 + ... + e(\phi_n) \times ⨍\phi_n$.

Proof. This proof is by using methods of Linear Programming in [AL75]. □

Example 4.1.8. Consider the same examples of CPL inference as before:

- $\{p_1, p_1 \to p_2\} \vDash_{\mathcal{B}} p_2$: $\ e(p_1) = e(p_1 \to p_2) = 1$

  Suppose that m is an EPPL model such that $[⨍p_1]_m = [⨍(p_1 \to p_2)]_m = 0.1$. Then, by Theorem 4.1.7, we have that $[⨍p_2]_m \leq 0.2$ (same result give by Theorem 4.1.3 because both premises are essential to the conclusion).

- $\{p_1, p_2, p_3, p_4\} \vDash_{\mathcal{B}}\ p_1 \wedge (p_2 \vee p_3)$:

  Suppose in this case that m is an EPPL model such that $[⨍p_i]_m = 0.1$ (with $i = 1, 2, 3, 4$). Then, the Essencialness Sum Theorem, for this CPL inference, give to the uncertainty of the conclusion that: $[⨍(p_1 \wedge (p_2 \vee p_3))]_m \leq 0.1 + 0.5 \times 0.1 + 0.5 \times 0.1 + 0 \times 0.1 = 0.2$, which is better than the upper bound found previously, because it eliminates some redundancy in the premises.

4.2 Interval Probability

This section aims to make this dissertation a little more multidisciplinary, showing how Linear Optimization concepts can be used in the context of probabilistic logics. First, we will consider one more abbreviation in EPPL, that represents that a real number r belongs to a real interval $[r_1, r_2]$, that is: $r \in [r_1, r_2] := (r_1 \leq r) \sqcap (r \leq r_2)$.

Clearly, given an arbitrary EPPL model m, we have that $m \vDash \big(r \in [r_1, r_2]\big)$ if and only if $m \vDash (r_1 \leq r)$ and $m \vDash (r \leq r_2)$.

The same notation will be used for the probabilistic operator, *i.e.* given ϕ a CPL formula, we have the same abbreviation: $\int\phi \in [r_1, r_2] := (r_1 \leq \int\phi) \sqcap (\int\phi \leq r_2)$.

In the following propositions we will consider that $k_1 = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2})$, $k_2 = \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)})$, $k_3 = \mathbf{P}(\Omega_{(\sim\phi_1)\wedge\phi_2})$ and $k_4 = \mathbf{P}(\Omega_{(\sim\phi_1)\wedge(\sim\phi_2)})$. Observe that:

$\mathbf{P}(\Omega) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)}) + \mathbf{P}(\Omega_{(\sim\phi_1)\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1)\wedge(\sim\phi_2)}) = 1$

$\Leftrightarrow k_1 + k_2 + k_3 + k_4 = 1.$

Proposition 4.2.1. $\vDash \big((\int\phi_1 = r_1) \sqcap (\int(\phi_1 \to \phi_2) = r_2)\big) \sqsupset \int\phi_2 \in [r_1 + r_2 - 1, r_2]$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. We want to prove that if $[\int\phi_1]_m = r_1$ and $[\int(\phi_1 \to \phi_2)]_m = r_2$, then $[\int\phi_2]_m \geq r_1 + r_2 - 1$ and $[\int\phi_2]_m \leq r_2$. Considering some results of previous chapter, we have that:

$[\int\phi_1]_m = \mathbf{P}(\Omega_{\phi_1}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)}) = k_1 + k_2$;
$[\int(\phi_1 \to \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1\to\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1)\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1)\wedge(\sim\phi_2)}) = k_1 + k_3 + k_4$;
$[\int\phi_2]_m = \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_2)\wedge\phi_1}) = k_1 + k_3$.

We want to solve the following linear optimization problem:

$$\begin{aligned} \text{min/max} \quad & k_1 + k_3 \\ \text{s.t.} \quad & k_1 + k_2 = r_1 \\ & k_1 + k_3 + k_4 = r_2 \\ & k_1 + k_2 + k_3 + k_4 = 1 \\ & k_1, k_2, k_3, k_4 \geq 0 \end{aligned} \tag{4.1}$$

$$\begin{cases} k_1 + k_2 = r_1 \\ k_1 + k_3 + k_4 = r_2 \\ k_1 + k_2 + k_3 + k_4 = 1 \end{cases} \Leftrightarrow \begin{cases} k_1 + k_2 = r_1 \\ k_1 + k_3 + k_4 = r_2 \\ k_3 + k_4 = 1 - r_1 \end{cases} \Leftrightarrow \begin{cases} k_1 + k_2 = r_1 \\ k_1 = r_1 + r_2 - 1 \\ k_3 + k_4 = 1 - r_1 \end{cases}$$
$$\Leftrightarrow \begin{cases} k_1 = r_1 + r_2 - 1 \\ k_2 = 1 - r_2 \\ k_3 = 1 - r_1 - k_4 \end{cases}$$

Together with nonnegative conditions, we have that:

$$\begin{cases} r_1 + r_2 - 1 \geq 0 \\ 1 - r_2 \geq 0 \\ 1 - r_1 - k_4 \geq 0 \\ k_4 \geq 0 \end{cases} \Leftrightarrow \begin{cases} r_1 + r_2 - 1 \geq 0 \\ r_2 \leq 1 \\ 1 - r_1 \geq 0 \\ 0 \leq k_4 \leq 1 - r_1 \end{cases} \Leftrightarrow \begin{cases} r_1 + r_2 \geq 1 \\ r_1 \leq 1 \\ r_2 \leq 1 \\ 0 \leq k_4 \leq 1 - r_1 \end{cases}$$

Then, $[\int\phi_2]_m = k_1 + k_3 = (r_1 + r_2 - 1) + (1 - r_1 - k_4) = r_2 - k_4 \;\Leftrightarrow\; k_4 = r_2 - [\int\phi_2]_m$.
Therefore, since $0 \leq k_4 \leq 1 - r_1$, we conclude that:

$0 \leq r_2 - [\int\phi_2]_m \leq 1 - r_1 \;\Leftrightarrow\; r_1 + r_2 - 1 \leq [\int\phi_2]_m \leq r_2.$

□

Proposition 4.2.2. $\models \big((\int\phi_1 = r_1) \sqcap (\int\phi_2 = r_2)\big) \sqsupset \int(\phi_1 \wedge \phi_2) \in [\max\{0, r_1+r_2-1\}, \min\{r_1, r_2\}]$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. We want to prove that if $[\int\phi_1]_m = r_1$ and $[\int\phi_2]_m = r_2$, then $[\int(\phi_1 \wedge \phi_2)]_m \geq \max\{0, r_1 + r_2 - 1\}$ and $[\int(\phi_1 \wedge \phi_2)]_m \leq \min\{r_1, r_2\}$.
$[\int\phi_1]_m = \mathbf{P}(\Omega_{\phi_1}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)}) = k_1 + k_2$;
$[\int\phi_2]_m = \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_2)\wedge\phi_1}) = k_1 + k_3$;
$[\int(\phi_1 \wedge \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) = k_1$.

The resulting linear optimization problem is as follows:

$$\begin{aligned} \text{min/max} \quad & k_1 \\ \text{s.t.} \quad & k_1 + k_2 = r_1 \\ & k_1 + k_3 = r_2 \\ & k_1 + k_2 + k_3 + k_4 = 1 \\ & k_1, k_2, k_3, k_4 \geq 0 \end{aligned} \tag{4.2}$$

$$\begin{cases} k_1 + k_2 = r_1 \\ k_1 + k_3 = r_2 \\ k_1 + k_2 + k_3 + k_4 = 1 \end{cases} \Leftrightarrow \begin{cases} k_2 = r_1 - k_1 \\ k_1 + k_3 = r_2 \\ k_3 + k_4 = 1 - r_1 \end{cases} \Leftrightarrow \begin{cases} k_2 = r_1 - k_1 \\ k_3 = r_2 - k_1 \\ k_4 = 1 - r_1 - r_2 + k_1 \end{cases}$$

Together with nonnegative conditions, we have that:

$$\begin{cases} k_1 \geq 0 \\ r_1 - k_1 \geq 0 \\ r_2 - k_1 \geq 0 \\ 1 - r_1 - r_2 + k_1 \geq 0 \end{cases} \Leftrightarrow \begin{cases} k_1 \geq 0 \\ k_1 \leq r_1 \\ k_1 \leq r_2 \\ k_1 \geq r_1 + r_2 - 1 \end{cases}$$

Therefore, since $[\int(\phi_1 \wedge \phi_2)] = k_1$, we conclude what we wanted to prove.

□

Proposition 4.2.3. $\models \big((\int\phi_1 = r_1) \sqcap (\int\phi_2 = r_2)\big) \sqsupset \int(\phi_1 \vee \phi_2) \in [\max\{r_1, r_2\}, \min\{1, r_1 + r_2\}]$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. We want to prove that if $[\int\phi_1]_m = r_1$ and $[\int\phi_2]_m = r_2$, then $[\int(\phi_1 \vee \phi_2)]_m \geq \max\{r_1, r_2\}$ and $[\int(\phi_1 \vee \phi_2)]_m \leq \min\{1, r_1 + r_2\}$.
$[\int\phi_1]_m = \mathbf{P}(\Omega_{\phi_1}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)}) = k_1 + k_2$;
$[\int\phi_2]_m = \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_2)\wedge\phi_1}) = k_1 + k_3$;
$[\int(\phi_1 \vee \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1\vee\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1)\wedge\phi_2}) + \mathbf{P}(\Omega_{\phi_1\wedge(\sim\phi_2)}) = k_1 + k_2 + k_3$.

$$\begin{aligned} \text{min/max} \quad & k_1 + k_2 + k_3 \\ \text{s.t.} \quad & k_1 + k_2 = r_1 \\ & k_1 + k_3 = r_2 \\ & k_1 + k_2 + k_3 + k_4 = 1 \\ & k_1, k_2, k_3, k_4 \geq 0 \end{aligned} \tag{4.3}$$

$$\begin{cases} k_1 + k_2 = r_1 \\ k_1 + k_3 = r_2 \\ k_1 + k_2 + k_3 + k_4 = 1 \end{cases} \Leftrightarrow \begin{cases} k_2 = r_1 - k_1 \\ k_1 + k_3 = r_2 \\ k_3 + k_4 = 1 - r_1 \end{cases} \Leftrightarrow \begin{cases} k_1 = r_1 + r_2 - 1 + k_4 \\ k_2 = 1 - r_2 - k_4 \\ k_3 = 1 - r_1 - k_4 \end{cases}$$

Together with nonnegative conditions, we have that:

$$\begin{cases} r_1 + r_2 - 1 + k_4 \geq 0 \\ 1 - r_2 - k_4 \geq 0 \\ 1 - r_1 - k_4 \geq 0 \\ k_4 \geq 0 \end{cases} \Leftrightarrow \begin{cases} k_4 \geq 1 - r_1 - r_2 \\ k_4 \leq 1 - r_2 \\ k_4 \leq 1 - r_1 \\ k_4 \geq 0 \end{cases}$$

Then, $[\int(\phi_1 \vee \phi_2]_m = k_1 + k_2 + k_3 = 1 - k_4 \Leftrightarrow k_4 = 1 - [\int(\phi_1 \vee \phi_2)]_m$, and since $\max\{0, 1 - r_1 - r_2\} \leq k_4 \leq \min\{1 - r_1, 1 - r_2\}$, we conclude that:

$$\begin{aligned} &\max\{0, 1 - r_1 - r_2\} \leq 1 - [\textstyle\int(\phi_1 \vee \phi_2)]_m \leq \min\{1 - r_1, 1 - r_2\} \\ &\Leftrightarrow \max\{-1, -r_1 - r_2\} \leq -[\textstyle\int(\phi_1 \vee \phi_2)]_m \leq \min\{-r_1, -r_2\} \\ &\Leftrightarrow \max\{r_1, r_2\} \leq [\textstyle\int(\phi_1 \vee \phi_2)]_m \leq \min\{1, r_1 + r_2\}. \end{aligned}$$

□

Proposition 4.2.4. $\vDash \big(\int(\phi_1 \to \phi_2) = r_1\big) \sqcap \big(\int(\phi_2 \to \phi_3) = r_2\big) \sqsupset \int(\phi_1 \to \phi_3) \in [r_1 + r_2 - 1, 1]$

Proof. This result was shown in a similar way to previous proposition, but now will have eight real variables k_i because, in this result, are involved three local formulas. □

These results are examples of how probabilities propagate in a CPL inference. However, the premises have a real value associated. Instead, it could be associated a range to each one. The following result ilustrates how probabilities are transmitted through *modus ponens*.

Theorem 4.2.5. $\vDash \big(\int\phi_1 \in [r_1, s_1]\big) \sqcap \big(\int(\phi_1 \to \phi_2) \in [r_2, s_2]\big) \sqsupset \int\phi_2 \in [r_1 + r_2 - 1, s_2]$

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. Suppose that $(r_1 \leq \int\phi_1) \sqcap (\int\phi_1 \leq s_1)$ and $\big(r_2 \leq \int(\phi_1 \to \phi_2)\big) \sqcap \big(\int(\phi_1 \to \phi_2) \leq s_2\big)$. As before, we have that:
$[\int\phi_1]_m = \mathbf{P}(\Omega_{\phi_1}) = \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2}) + \mathbf{P}(\Omega_{\phi_1 \wedge (\sim\phi_2)}) = k_1 + k_2$;
$[\int(\phi_1 \to \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1 \to \phi_2}) = \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1) \wedge \phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_1) \wedge (\sim\phi_2)}) = k_1 + k_3 + k_4$;
$[\int\phi_2]_m = \mathbf{P}(\Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1 \wedge \phi_2}) + \mathbf{P}(\Omega_{(\sim\phi_2) \wedge \phi_1}) = k_1 + k_3$.

We need to solve the following linear optimization problems:

$$\begin{aligned} \text{min/max} \quad & k_1 + k_3 \\ \text{s.t.} \quad & r_1 \leq k_1 + k_2 \leq s_1 \\ & r_2 \leq k_1 + k_3 + k_4 \leq s_2 \\ & k_1 + k_2 + k_3 + k_4 = 1 \\ & k_1, k_2, k_3, k_4 \geq 0 \end{aligned} \tag{4.4}$$

□

Corollary 4.2.6. $\vDash (\fint\phi_1 \leq \varepsilon_1) \sqcap \big(\fint(\phi_1 \to \phi_2) \leq \varepsilon_2\big) \sqsupset \fint\phi_2 \leq \varepsilon_1 + \varepsilon_2$

Proof. We only need to observe that $\int\phi \in [1-\varepsilon, 1]$ is the same as $(0 \leq \fint\phi) \sqcap (\fint\phi \leq \varepsilon)$. $\square$

Note that the generated linear optimization problems become increasingly complicated, and therefore it would be more interesting to use computational tools to solve them.

The following Theorem was originally written in the context of a probabilistic propositional logic with a slightly different syntax, and is adapted here to EPPL. It is a generalization of the previous propositions, that is, the construction of the Linear Optimization problem described in this theorem is already illustrated in the propositions.

Theorem 4.2.7. ([Hai96]) Let $\{\phi_1, ..., \phi_n\} \vDash_{\mathcal{B}} \phi$ be a valid CPL inference, and r_i, s_i real algebraic numbers such that $0 \leq r_i \leq s_i \leq 1$ $(i = 1, ..., n)$. Then,

$$\vDash \big(\int\phi_1 \in [r_1, s_1] \sqcap \int\phi_2 \in [r_2, s_2] \sqcap ... \sqcap \int\phi_n \in [r_n, s_n]\big) \sqsupset \int\phi \in [r_{\mathrm{LB}}, s_{\mathrm{UB}}]$$

where the optimal set $[r_{\mathrm{LB}}, s_{\mathrm{UB}}]$ is a real interval whose end points are the minimum and maximum (respectively) of a linear optimization problem.

Proof. Theorem 4.61 in [Hai96] (the idea of this proof is below).

For any ϕ_i, consider that $K_{(\phi_i)}$ the set of its constituients k_j $(j = 1, ..., 2^n)$. The linear programming problems that we need to solve in order to find the interval $[r_{\mathrm{LB}}, s_{\mathrm{UB}}]$ are the following:

$$\begin{aligned}
\text{min/max} \quad & z = \sum_{k_j \in K_{(\phi)}} k_j \\
\text{s.t.} \quad & r_i \leq \sum_{k_j \in K_{(\phi_i)}} k_j \leq s_i \ , \quad i = 1, ..., n \\
& \sum_{j=1}^{2^n} k_j \ = \ 1 \\
& k_j \geq 0 \ , \quad i = 1, ..., 2^n
\end{aligned} \tag{4.5}$$

To conclude this proof, the idea is to show that no feasible point can lie outside the interval $[\min z, \max z]$ (as shown in [Hai96]).

$\square$

4.3 Conditional

In Probability Theory, conditional probability refers to the probability of an event E_2 given that another event E_1 has occurred. Usually, we denote as $\mathbf{P}(E_2|E_1)$, and it is read as *probability of E_2 given E_1*.

In this section, we aim at adapting this probabilistic concept into EPPL. In order to not change neither local logic (CPL) or global logic (EPPL), we will introduce the conditional probability as a real probabilistic term, with the syntax $\int(\phi_2|\phi_1)$.

In a semantic context, given $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ an arbitrary EPPL model, the interpretation of a condicional term is defined as in probability theory, as follows:

$$[\int(\phi_2|\phi_1)]_m = \frac{[\int(\phi_1 \wedge \phi_2)]_m}{[\int\phi_1]_m}, \text{ if } [\int\phi_1]_m \neq 0; \text{ and } [\int(\phi_2|\phi_1)]_m = 1 \text{ otherwise.}$$

Moreover, in order to make EPPL with conditional terms still weakly sound and complete, we will consider the same Hilbert calculus system (defined in Table 3.3), with the addition of these two axioms:

[C0] $\vdash (\int\phi_1 = 0) \sqsupset \big(\int(\phi_2|\phi_1) = 1\big)$
[C1] $\vdash \int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int(\phi_2|\phi_1)$

The next result is important because it proves that the defined conditional axioms are well-defined (taking into account the conditional semantics), and for example proves that the chain rule (in Proposition 4.3.2) is also valid in a semantical context.

Theorem 4.3.1. This Hilbert calculus for EPPL with conditionals is weakly sound.

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an arbitrary EPPL model. Whereas that it has been proved the soundness of EPPL, we need to prove that both conditional axioms are semantically valid:

[C0] $\vdash \big(\int\phi_1 = 0\big) \sqsupset (\int(\phi_2|\phi_1) = 1)$:
$m \vDash (\int\phi_1 = 0) \sqsupset \big(\int(\phi_2|\phi_1) = 1\big)$ iff $m \nvDash \int\phi_1 = 0$ or $m \vDash \int(\phi_2|\phi_1) = 1$
iff $[\int\phi_1]_m \neq 0$ or $[\int(\phi_2|\phi_1)]_m = 1$.
If $[\int\phi_1]_m \neq 0$ it is done. Otherwise, *i.e.* if $[\int\phi_1]_m = 0$, then follows immediatly that $[\int(\phi_2|\phi_1)]_m = 1$ by definition.

[C1] $\vdash \int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int(\phi_2|\phi_1)$:
$m \vDash \int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int(\phi_2|\phi_1)$ iff $[\int(\phi_1 \wedge \phi_2)]_m = [\int\phi_1]_m \times [\int(\phi_2|\phi_1)]_m$.
If $[\int\phi_1]_m = 0$, then $[\int(\phi_1 \wedge \phi_2)]_m = 0$, because $0 \leq [\int(\phi_1 \wedge \phi_2)]_m \leq [\int\phi_1]_m = 0$ since $\Omega_{\phi_1 \wedge \phi_2} \subseteq \Omega_{\phi_1}$. Otherwise, that is, if $[\int\phi_1]_m \neq 0$, then we have that:

$$[\textstyle\int(\phi_1 \wedge \phi_2)]_m = [\int\phi_1]_m \times \dfrac{[\int(\phi_1 \wedge \phi_2)]_m}{[\int\phi_1]_m} \ ,$$

which is easily checked to be true by real properties, and the result is proved.

□

In the study of conditionals in Probability Theory, one of the most known and used rules is the called *Chain Rule*. The next proposition fits this rule into EPPL conditional formulas as defined here.

Proposition 4.3.2. Given $\phi_1, ..., \phi_n \in Form(\mathcal{B})$ (with $n \geq 2$), we have that:
[CR] $\vdash \int(\phi_1 \wedge ... \wedge \phi_n) = \int\phi_1 \times \int(\phi_2|\phi_1) \times \int\big(\phi_3|(\phi_1 \wedge \phi_2)\big) \times ... \times \int\big(\phi_n|(\phi_1 \wedge ... \wedge \phi_{n-1})\big)$

Proof. We will prove this result using induction over natural numbers.
Base step: If $n = 2$, the result is proved directly by **[C1]**.
Inductive step: Suppose that this result is verified for $n{=}k{-}1$. Then:

1.	$\vdash \int(\phi_1 \wedge ... \wedge \phi_{n-1}) = \int\phi_1 \times \int(\phi_2|\phi_1) \times ... \times \int\big(\phi_{n-1}|(\phi_1 \wedge ... \wedge \phi_{n-2})\big)$	**[Ind-Hyp]**
2.	$\vdash \int((\phi_1 \wedge ... \wedge \phi_{n-1}) \wedge \phi_n) = \int(\phi_1 \wedge ... \wedge \phi_{n-1}) \times \int\big(\phi_n|(\phi_1 \wedge ... \wedge \phi_{n-1})\big)$	**[C1]**
3.	$\vdash \int(\phi_1 \wedge ... \wedge \phi_n) = \int\phi_1 \times \int(\phi_2|\phi_1) \times ... \times \int\big(\phi_n|(\phi_1 \wedge ... \wedge \phi_{n-1})\big)$	**[Real(1,2)]**

□

Note that in Probability Theory there is no notion of logical implication, that is, when we referred to the probability of *if* $E1$ *then* $E2$, we associate to this the conditional probability $\mathbf{P}(E_2|E_1)$. However, in a probabilistic logic with conditional, it is clear that $\int(\phi_1 \to \phi_2)$ and $\int(\phi_2|\phi_1)$ define completely different probabilities.

Given a sentence *if* ϕ_1 *then* ϕ_2, there is no agreement in the literature in which one of the two previously probabilities associated to it. In a logical context, it is more intuitive to associate $\int(\phi_1 \to \phi_2)$; but in a probability context, usually it is associated to the conditional $\int(\phi_2|\phi_1)$. The following example shows to us that, in general, they are completely different.

Example 4.3.3. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be the EPPL model that represent a roll-dice (as in Example 3.2.2). Consider that p_1 represents *outcome is even*, and p_2 represents *outcome is a prime number*. That is, $\Omega_{p_1} = \{2, 4, 6\}$ and $\Omega_{p_2} = \{2, 3, 5\}$. Then:

- $[\int(p_1 \to p_2)]_m = \mathbf{P}(\Omega_{\sim p_1} \cup \Omega_{p_2}) = \mathbf{P}(\{1, 3, 5\} \cup \{2, 3, 5\}) = \mathbf{P}(\{1, 2, 3, 5\}) = \frac{2}{3}$;
- $[\int(p_2|p_1)]_m = \frac{[\int(p_1 \wedge p_2)]_m}{[\int(p_1)]_m} = \frac{\mathbf{P}(\Omega_{p_1} \cap \Omega_{p_2})}{\mathbf{P}(\Omega_{p_1})} = \frac{\mathbf{P}(\{2\})}{\mathbf{P}(\{2,4,6\})} = \frac{1}{3}$.

The following theorem give us a sufficient and necessary condition to $\int(\phi_1 \to \phi_2)$ and $\int(\phi_2|\phi_1)$ be equal probabilities.

Theorem 4.3.4. $\vDash \big(\int(\phi_1 \to \phi_2) = \int(\phi_2|\phi_1)\big) \equiv \big(\int\phi_1 = 1 \sqcup \int(\phi_1 \to \phi_2) = 1\big)$.

Proof. We need to show that, given an arbitrary EPPL model $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$,

$m \vDash \int(\phi_1 \to \phi_2) = \int(\phi_2|\phi_1)$ if and only if $m \vDash \int\phi_1 = 1 \sqcup \int(\phi_1 \to \phi_2) = 1$

Considering the definition of $[\int(\phi_2|\phi_1)]_m$, we will divide the proof into two cases:

If $[\int\phi_1]_m = 0$, then $[\int(\phi_2|\phi_1)]_m = 1$ always by definition; and $[\int(\phi_1 \to \phi_2)]_m = \mathbf{P}(\Omega_{\phi_1\to\phi_2})$ $= \mathbf{P}(\Omega_{\sim\phi_1} \cup \Omega_{\phi_2}) \geq \mathbf{P}(\Omega_{\sim\phi_1}) = 1 - \mathbf{P}(\Omega_{\phi_1}) = 1 - 0 = 1$. Then $[\int(\phi_1 \to \phi_2)]_m = 1$.

Otherwise, that is, if $[\int\phi_1]_m \neq 0$, then:

$m \vDash \int(\phi_1 \to \phi_2) = \int(\phi_2|\phi_1)$ iff $[\int(\phi_1 \to \phi_2)]_m = [\int(\phi_2|\phi_1)]_m$
iff $\mathbf{P}(\Omega_{\phi_1\to\phi_2}) = \mathbf{P}(\Omega_{\phi_1\wedge\phi_2})/\mathbf{P}(\Omega_{\phi_1})$ iff $\mathbf{P}(\Omega_{\sim\phi_1\vee\phi_2}) = \big(\mathbf{P}(\Omega_{\phi_1}) - \mathbf{P}(\Omega_{\phi_1\wedge\sim\phi_2})\big)/\mathbf{P}(\Omega_{\phi_1})$
iff $1 - \mathbf{P}(\Omega_{\phi_1\wedge\sim\phi_2}) = 1 - \mathbf{P}(\Omega_{\phi_1\wedge\sim\phi_2})/\mathbf{P}(\Omega_{\phi_1})$ iff $\mathbf{P}(\Omega_{\phi_1}) = 1$ or $\mathbf{P}(\Omega_{\phi_1\wedge\sim\phi_2}) = 0$
iff $\mathbf{P}(\Omega_{\phi_1}) = 1$ or $\mathbf{P}(\Omega_{\phi_1\to\phi_2}) = 1$ iff $[\int\phi_1]_m = 1$ or $[\int(\phi_1 \to \phi_2)]_m = 1$
iff $m \vDash \int\phi_1 = 1 \sqcup \int(\phi_1 \to \phi_2) = 1$.

□

Furthermore, we can prove that the probability of CPL implication is one if and only if the probability of the conditional is also one and, despite the differences between them, this two concepts are related in a way.

Theorem 4.3.5. $\vDash \big(\int(\phi_2|\phi_1) = 1\big) \equiv \big(\int(\phi_1 \to \phi_2) = 1\big)$.

Proof. Let $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ be an EPPL model. We want to prove that $[\int(\phi_1 \to \phi_2)]_m = 1$ if and only if $[\int(\phi_2|\phi_1)]_m = 1$. Considering the semantical definition of $[\int(\phi_2|\phi_1)]_m$, we will divide this prove into two cases:

If $[\int\phi_1]_m = 0$, it is already proved in the previous theorem.

Otherwise, that is, if $[\int\phi_1]_m \neq 0$, then:
$[\int(\phi_2|\phi_1)]_m = 1$ iff $[\int(\phi_1 \wedge \phi_2)]_m/[\int(\phi_1)]_m = 1$ iff $[\int(\phi_1 \wedge \phi_2)]_m = [\int(\phi_1)]_m$
iff $\mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1})$ iff $\mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2}) = \mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\phi_2}) + \mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\sim\phi_2})$
iff $\mathbf{P}(\Omega_{\phi_1} \cap \Omega_{\sim\phi_2}) = 0$ iff $\mathbf{P}\big(\Omega \setminus (\Omega_{\phi_1} \cap \Omega_{\sim\phi_2})\big) = 1$ iff $\mathbf{P}(\Omega_{\sim\phi_1} \cup \Omega_{\phi_2}) = 1$
iff $\mathbf{P}(\Omega_{(\sim\phi_1)\vee\phi_2}) = 1$ iff $\mathbf{P}(\Omega_{\phi_1\to\phi_2}) = 1$ iff $[\int(\phi_1 \to \phi_2)]_m = 1$.

□

4.3.1 Independence of Formulas

In Probability Theory, we say that two events are (statistically) independents if the occurrence of one does not affect the probability of the other. This section aims to bring into EPPL a similar independence concept.

The formal definition in Probability Theory is that two events E_1 and E_2 are independent if $\mathbf{P}(E_1 \cap E_2) = \mathbf{P}(E_1) \times \mathbf{P}(E_2)$. Given that, we will use this idea to introduce independence of formulas in EPPL.

Syntactically, the global formula $(\phi_1 \perp\!\!\!\perp \phi_2)$ will represent that the CPL formulas ϕ_1 and ϕ_2 are independent. We will introduce it in EPPL semantics and Hilbert calculus in a very standard way. Given an EPPL model $m = (\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$, we have that:

$m \vDash (\phi_1 \perp\!\!\!\perp \phi_2)$ iff $[\int(\phi_1 \wedge \phi_2)]_m = [\int\phi_1]_m \times [\int\phi_2]_m$.

Considerer the EPPL Hilbert calculus in Table 3.3, we will be adding a new axiom to it:

[Indep] $\vdash (\phi_1 \perp\!\!\!\perp \phi_2) \equiv \big(\int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int\phi_2\big)$

In some articles (*e.g.* [BM09] and [BMN10]) this feature is considered as belonging to EPPL syntax. However, the global formula $(\phi_1 \perp\!\!\!\perp \phi_2)$ is semantically equivalent to the following EPPL formula: $\big(\int(\phi_1 \wedge \phi_2) \leq \int\phi_1 \times \int\phi_2\big) \sqcap \big(\int\phi_1 \times \int\phi_2 \leq \int(\phi_1 \wedge \phi_2)\big)$.

Because of that, we decide not to include the independence formulas in the definition of EPPL, in order to initially have the simplest possible probabilistic propositional logic, and then adding other probabilistic concepts in that logic.

Now we want to analize the existing duality with Probability Theory in the relations between independence and conditionals. In [IG10], the concept of conditional independence in probabilistic logics is studied, and here we want only to show that we can write the independence of formulas in terms of conditional formulas (as it happens in probability theory).

When the probability of the event that we are conditioning is not zero, the notion of independence can be rewritten with conditionals. We can also rewrite the notion of independent formulas using formulas with conditional, as shown in the following theorem.

Theorem 4.3.6. **[CI]** $(\int\phi_1 > 0) \vdash (\phi_1 \perp\!\!\!\perp \phi_2) \equiv \big(\int(\phi_2|\phi_1) = \int\phi_2\big)$

Proof. Considerer the definition of $\equiv$ and Theorems 3.3.3 and 3.3.4, prove this theorem is equivalent to demonstrate the following two results:

(i) $(\int\phi_1 > 0), (\phi_1 \perp\!\!\!\perp \phi_2) \vdash \int(\phi_2|\phi_1) = \int\phi_2$:

1.	$\int\phi_1 > 0$	**[Hyp]**
2.	$(\phi_1 \perp\!\!\!\perp \phi_2)$	**[Hyp]**
3.	$(\phi_1 \perp\!\!\!\perp \phi_2) \sqsupset \big(\int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int\phi_2\big)$	**[Indep]**
4.	$\int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int\phi_2$	**[GMP(2,3)]**
5.	$\int(\phi_1 \wedge \phi_2) = \int\phi_1 \times \int(\phi_2|\phi_1)$	**[C1]**
6.	$\int(\phi_2|\phi_1) = \phi_2$	**[Real(1,4,5)]**

(ii) $(\int \phi_1 > 0), (\int(\phi_2|\phi_1) = \int \phi_2) \vdash (\phi_1 \perp\!\!\!\perp \phi_2)$

1.	$\int(\phi_2\|\phi_1) = \phi_2$	**[Hyp]**
2.	$\int(\phi_1 \wedge \phi_2) = \int \phi_1 \times \int(\phi_2\|\phi_1)$	**[C1]**
3.	$\int(\phi_1 \wedge \phi_2) = \int \phi_1 \times \int \phi_2$	**[Real(1,2)]**
4.	$(\int(\phi_1 \wedge \phi_2) = \int \phi_1 \times \int \phi_2) \sqsupset (\phi_1 \perp\!\!\!\perp \phi_2)$	**[Indep]**
5.	$(\phi_1 \perp\!\!\!\perp \phi_2)$	**[GMP(4,5)]**

□

4.3.2 Suppositional Logic

Instead of putting conditional on our probabilistic logic as a real term, alternatively, we can change the basic logic (CPL) to a new logic where the conditional formulas belong to basic logic syntax - at this new local logic we will call *suppositional logic* (denoted as $\mathcal{S}$).

$$\psi \ := \ \phi \mid (\phi|\phi) \ , \quad \phi \in Form(\mathcal{B})$$

Table 4.1: Suppositional Logic Syntax

We will start by defining a semantics for this logic. The idea is to make an extension of the CPL concepts (of valuations, semantical consequences and tautologies) for this new language, in the sense that if we do not have conditional formulas, this semantics coincide with the CPL satisfaction system (defined in Subsection 2.2.2).

Definition 4.3.7. An **u-valuation** is a mapping $v : Var(\mathcal{B}) \to \{0,1\}$ $(\subset \{0, \mathrm{u}, 1\})$, that can be naturaly extended for $\bar{v} : Form(\mathcal{S}) \to \{0, u, 1\}$ by $\bar{v}(p) = v(p)$ if $p \in Var(\mathcal{B})$, $\bar{v}(\sim\phi) = 1 - \bar{v}(\phi)$, $\bar{v}(\phi_1 \to \phi_2) = \max\{1 - \bar{v}(\phi_1), \bar{v}(\phi_2)\}$ and $\bar{v}(\phi_2|\phi_1) = \mathrm{u}$ if $\bar{v}(\phi_1) = 0$ and $\bar{v}(\phi_2|\phi_1) = \bar{v}(\phi_2)$ otherwise.

As it happens in CPL, we will abbreviate $\bar{v}$ to v. Note that an u-valuation of a formula without conditional behaves as in CPL, that is, $v(\phi) \in \{0,1\}$. In the following definitions we will introduce a satisfaction relation for this logic.

Definition 4.3.8. Given an u-valuation v and a set of suppositional formulas $\Psi = \{\psi_1, ..., \psi_n\}$, we will say that v **confirms** Ψ if and only if:

- $v(\psi_i) \neq 0$, for all $i \in \{1, ..., n\}$;
- and exists $j \in \{1, ..., n\}$ such that $v(\psi_j) = 1$.

Moreover, we say that v **falsifies** Ψ if and only if $v(\psi_j) = 0$, for some $j \in \{1, ..., n\}$.

Definition 4.3.9. Let $\Psi = \{\psi_1, ..., \psi_n\}$ be a set of suppositional formulas (premises) and ψ other suppositional formula (conclusion). We say that ψ is a **semantical consequence** of $\Psi = \{\psi_1, ..., \psi_n\}$ $(\Psi \vDash_{\mathcal{S}} \psi)$ if and only if both of following conditions are satisfied:

- $v(\psi) \neq 0$, for all u-valuations v that do not falsify Ψ;
- there exists an u-valuation that confirms Ψ and $v(\psi) = 1$.

Moreover, we say that ψ is **u-valid** (denoted as $\vDash_{\mathcal{S}} \psi$) if and only if $v(\psi) \neq 0$ for all u-valuation v, and exists an u-valuation v such that $v(\psi) = 1$.

This satisfaction system for supposicional logic results of a combination between the semantics presented for conditional formula in [Hai96], which is based on the existence of a third truth value u, and semantics presented in [Ada98] based only in truth tables.

Lemma 4.3.10. A formula $\phi \in Form(\mathcal{B})$ (that is, without conditional occurences) is u-valid if and only if is valid in CPL.

Proof. Straightforward, by definition of CPL valuation and u-valuation to formulas without conditionals (that is, they are defined in the same way). □

The idea is to define an axiomatic system for this logic (Table 4.2), so that we can at least establish weak soundness. The proposed system for this conditional logic is adapted for an existing inference system in [Ada98], together with the axiomatization of CPL previously defined in Table 2.2. Note that, in the following table, ϕ_1, ϕ_2 and ϕ_3 can not be conditional formulas, but always be CPL formulas.

Axioms:

[Ax1]	$\vdash_{\mathcal{S}} \ \phi_1 \to (\phi_2 \to \phi_1)$
[Ax2]	$\vdash_{\mathcal{S}} \ (\phi_1 \to (\phi_2 \to \phi_3) \to ((\phi_1 \to \phi_2) \to (\phi_1 \to \phi_3))$
[Ax3]	$\vdash_{\mathcal{S}} \ (\sim\phi_2 \to \sim\phi_1) \to ((\sim\phi_2 \to \phi_1) \to \phi_2)$

Inference Rules:

[MP]	$\phi_1, (\phi_1 \to \phi_2) \ \vdash_{\mathcal{S}} \ \phi_2$
[LC]	$(\phi_1 \to \phi_2) \ \vdash_{\mathcal{S}} \ \phi_2 \vert \phi_1$
[CE1]	$\phi_1 \ \vdash_{\mathcal{S}} \ \phi_1 \vert \top$
[CE2]	$\phi_1 \vert \top \ \vdash_{\mathcal{S}} \ \phi_1$
[EA]	$\phi_1 \leftrightarrow \phi_2, \ \phi_3 \vert \phi_1 \ \vdash_{\mathcal{S}} \ \phi_3 \vert \phi_2$
[DA]	$\phi_3 \vert \phi_1, \ \phi_3 \vert \phi_2 \ \vdash_{\mathcal{S}} \ \phi_3 \vert (\phi_1 \vee \phi_2)$
[RT]	$\phi_2 \vert \phi_1, \ \phi_3 \vert (\phi_1 \wedge \phi_2) \ \vdash_{\mathcal{S}} \ \phi_3 \vert \phi_1$
[AR]	$\phi_2 \vert \phi_1, \ \phi_3 \vert \phi_1 \ \vdash_{\mathcal{S}} \ \phi_3 \vert (\phi_1 \wedge \phi_2)$

Table 4.2: Suppositional Deductive System

In the following theorem we will show that suppositional logic is sound, *i.e.* this means that this conditional inference rules are valid in sense that each conclusion is a semantical consequence of their respective premises (see Definition 4.3.9).

Theorem 4.3.11. Suppositional Logic is sound.

Proof. We need to show that all axioms and inference rules in Table 4.3.2 are semantically valid in suppositional logic (considering all previous definitions).

First, by Lemma 4.3.10, the axioms **[Ax1]**, **[Ax2]**, **[Ax3]** and the inference rule **[MP]** are already sound, because these are instances of axioms and inference rule of CPL, and the latter is already sound.

All other inference rules will be proved using truth-tables for suposicional logic (that is, exploring each one of all possible u-valuations).

[LC] $(\phi_1 \to \phi_2) \vdash_{\mathcal{S}} \phi_2|\phi_1$:

ϕ_1	ϕ_2	$\phi_1 \to \phi_2$	$\phi_1\|\phi_2$	
1	1	**1**	**1**	✓
1	0	0	0	
0	1	**1**	**u**	
0	0	**1**	**u**	

[CE1] $\phi_1 \vdash_{\mathcal{S}} \phi_1|\top$ and **[CE2]** $\phi_1|\top \vdash_{\mathcal{S}} \phi_1$:

$\top$	ϕ_1	$\phi_1\|\top$	
1	**1**	**1**	✓
1	0	0	

[EA] $\phi_1 \leftrightarrow \phi_2,\ \phi_3|\phi_1 \vdash_{\mathcal{S}} \phi_3|\phi_2$:

ϕ_1	ϕ_2	ϕ_3	$\phi_1 \leftrightarrow \phi_2$	$\phi_3\|\phi_1$	$\phi_3\|\phi_2$	
1	1	1	**1**	**1**	**1**	✓
1	1	0	1	0	0	
1	0	1	0	1	u	
1	0	0	0	0	u	
0	1	1	0	u	1	
0	1	0	0	u	0	
0	0	1	**1**	**u**	**u**	
0	0	0	**1**	**u**	**u**	

[DA] $\phi_3|\phi_1,\ \phi_3|\phi_2 \vdash_{\mathcal{S}} \phi_3|(\phi_1 \vee \phi_2)$:

ϕ_1	ϕ_2	ϕ_3	$\phi_3\|\phi_1$	$\phi_3\|\phi_2$	$\phi_3\|(\phi_1 \vee \phi_2)$	
1	1	1	**1**	**1**	**1**	✓
1	1	0	0	0	0	
1	0	1	**1**	**u**	**1**	✓
1	0	0	0	u	0	
0	1	1	**u**	**1**	**1**	✓
0	1	0	u	0	0	
0	0	1	**u**	**u**	**u**	
0	0	0	**u**	**u**	**u**	

[RT] $\phi_2|\phi_1,\ \phi_3|(\phi_1 \wedge \phi_2)\ \vdash_{\mathcal{S}}\ \phi_3|\phi_1$:

ϕ_1	ϕ_2	ϕ_3	$\phi_2\|\phi_1$	$\phi_3\|(\phi_1\wedge\phi_2)$	$\phi_3\|\phi_1$	
1	1	1	**1**	**1**	**1**	✓
1	1	0	1	0	0	
1	0	1	0	u	1	
1	0	0	0	u	0	
0	1	1	**u**	**u**	**u**	
0	1	0	**u**	**u**	**u**	
0	0	1	**u**	**u**	**u**	
0	0	0	**u**	**u**	**u**	

[AR] $\phi_2|\phi_1,\ \phi_3|\phi_1\ \vdash_{\mathcal{S}}\ \phi_3|(\phi_1 \wedge \phi_2)$:

ϕ_1	ϕ_2	ϕ_3	$\phi_2\|\phi_1$	$\phi_3\|\phi_1$	$\phi_3\|(\phi_1\wedge\phi_2)$	
1	1	1	**1**	**1**	**1**	✓
1	1	0	1	0	0	
1	0	1	0	1	u	
1	0	0	0	0	u	
0	1	1	**u**	**u**	**u**	
0	1	0	**u**	**u**	**u**	
0	0	1	**u**	**u**	**u**	
0	0	0	**u**	**u**	**u**	

□

In order to exemplify how this deductive system works, we will prove a derived inference rule that corresponds to a kind of *modus ponens* for conditional formulas.

Proposition 4.3.12. **[CMP]** $\phi_1,\ \phi_2|\phi_1\ \vdash_{\mathcal{S}}\ \phi_2$

Proof.

1.	ϕ_1	**[Hyp]**
2.	$\phi_2\|\phi_1$	**[Hyp]**
3.	$\phi_1\|\top$	**[CE1(1)]**
4.	$\phi_1 \leftrightarrow (\top \wedge \phi_1)$	**[Taut]**
5.	$\phi_2\|(\top \wedge \phi_1)$	**[EA(4,2)]**
6.	$\phi_2\|\top$	**[RT(3,5)]**
7.	ϕ_2	**[CE2(6)]**

□

We can easily see that this derived inference rule is semantical valid, *i.e.* there is just one u-valuation that confirms $\{\phi_1, \phi_2|\phi_1\}$, and this valuation gives 1 to the conclusion ϕ_2.

ϕ_1	$\phi_2\|\phi_1$	ϕ_2	
1	**1**	**1**	✓
1	0	0	
0	u	1	
0	u	0	

Remember that the idea would be to replace in EPPL (defined in Chapter 3) our local logic (*i.e.* CPL) by suppositional logic - this new probabilistic logic we will designate by EPPLC (EPPL with *Conditional*). For this, first we will need to prove that suppositional logic is also a complete logic system according to its semantics. In the following table lies the syntax of this new logic, that is almost equal to the EPPL syntax, changing only in the basic formulas.

Local Conditional Formulas:

$$\psi \ := \ \phi \mid (\phi|\phi) \ , \quad \phi \in \mathit{Form}(\mathcal{B})$$

Real Probabilistic Terms:

$$t \ := \ x \mid 0 \mid 1 \mid \textstyle\int\psi \mid (t+t) \mid (t \times t) \ , \ x \in \mathit{Var}(\mathbb{R})$$

Global Formulas:

$$\delta \ := \ \Box\psi \mid (t \leq t) \mid (\neg\delta) \mid (\delta \sqsupset \delta) \ .$$

Table 4.3: EPPLC Syntax

With this EPPLC syntax, formulas like $\Box(\phi_1|\phi_1)$ and $\int(\phi_2|\phi_1) \leq \int(\phi_3)$ were defined in this new probabilistic logic, but not formulas like $\Box\big(\phi_1|(\phi_2|\phi_1)\big)$. Now that we set a syntax to EPPLC, the next steps would be define a semantic and a Hilbert calculus for EPPLC (which would be identical to the defined previously in sections 3.2 and 3.3 for EPPL), and study some properties of this logic, namely its Soundness and Completeness.

4.4 Probabilization of Logic Systems

In the previous chapter, we worked in one probabilization of the CPL. Now, our interest is to study a way to generalize this probabilization process, *i.e.* we want to make probabilistic any arbitrary logic system.

In general, a logic system aims at formally reasoning about a wide range of entities such as actions, knowledge, belief, probabilities, among many others. When we want to reason about to completely different entities, it is evident that we need of combine two (or more) logic systems. The combination of Logics is a topic widely studied in Logic (*e.g.* [She73] and [SSC05]). Combining logics consists in the combination of two (or more) logic systems (or satisfaction systems) to a single logic system (or a satisfaction system), using some kind of technique. An example for that is fibring logics ([CSS05]), that consists in merging two (or more) axiomatic systems into a new one with the axioms and inference rules of both systems. Alternatively, we can use asymmetric combinations of logics ([RSS13]) that consists in defining one of the systems in a higher level than the other, keeping the first unchanged. Recently, some scientifical works ([MNM16], [MNBM16]) have been done on context of the dynamisation and hybridization process, which are examples of asymmetric combinations. The process of probabilization of logics is very similar to both of these processes, but in this we put probabilities in a higher level (creating a new probabilistic logic).

Given a logic system, it is understood by probabilization the enrichment of this system with probability features. This section defines an operator that combines any satisfaction system with a probabilistic propositional logic (see [Bal10] for more details in the process) in two distict levels (as it happens in EPPL). This operator uses an exogenous approach, *i.e.* we fix the satisfaction system that we want to probabilize (keeping its syntax and semantics unchanged), and introduce probabilistic formulas at a higher level.

Definition 4.4.1. Let $\mathscr{L} = (L, \mathcal{M}, \vDash)$ be any satisfaction system. The **probabilization operator** transforms $\mathscr{L}$ in the system $\mathscr{L}^p = (L^p, \mathcal{M}^p, \vDash^p)$, defined as follows:

- L^p is defined by:

$$t := \ r \mid (\textstyle\int\phi) \mid (t+t) \mid (t \times t)$$
$$\delta := \ (t \leq t) \mid (\neg\delta) \mid (\delta \sqsupset \delta) \ , \qquad \text{with } \phi \in \text{Form}(\mathscr{L}) \text{ and } r \in \text{Alg}(\mathbb{R}).$$

- $\mathcal{M}^p$ is the class of all tuples $m = (\Omega, \mathcal{F}, \mathbf{P}, V)$, where $(\Omega, \mathcal{F}, \mathbf{P})$ is a probability space, and $V : \Omega \to \mathcal{M}$ is a *measurable valuation*, in the sense that for all $\phi \in Form(\mathscr{L})$, we have that $V^{-1}(\phi) := \{\omega \in \Omega : V(\omega) \vDash \phi\} \in \mathcal{F}$.

- the satisfaction relation $\vDash^p$ is recursively defined as:

  - $[r]_m := r$, for each $r \in Alg(\mathbb{R})$;
  - $[\int\phi]_m = \mathbf{P}\big(V^{-1}(\phi)\big)$;
  - $[t_1 + t_2]_m := [t_1]_m + [t_2]_m$;
  - $[t_1 \times t_2]_m := [t_1]_m \times [t_2]_m$;

  - $m \vDash^p (t_1 \leq t_2)$ iff $[t_1]_m \leq [t_2]_m$;
  - $m \vDash^p (\neg\delta)$ iff $m \nvDash^p \delta$;
  - $m \vDash^p (\delta_1 \sqsupset \delta_2)$ iff ($m \nvDash^p \delta_1$ or $m \vDash^p \delta_2$),

  with $m \in \mathcal{M}^p$ and $\delta \in Form(\mathscr{L}^p)$.

4.4.1 Probabilization of Classical Propositional Logic

Let $\mathscr{L} = (L, \mathcal{M}, \vDash)$ be CPL satisfaction system (as defined in subsection 2.2.2). The probabilization operator defined previoulsy transforms CPL in the satisfaction probabilistic system $\mathscr{L}^p = (L^p, \mathcal{M}^p, \vDash^p)$, defined by:

- L^p is defined by:

$$t := \ r \mid (\textstyle\int \phi) \mid (t + t) \mid (t \times t)$$
$$\delta := \ (t \leq t) \mid (\neg \delta) \mid (\delta \sqsupset \delta) \ , \qquad \text{with } \phi \in \text{Form}(\mathscr{L}) \text{ and } r \in \text{Alg}(\mathbb{R}).$$

- $\mathcal{M}^p$ is the class of all models $m = (\Omega, \mathcal{F}, \mathbf{P}, V)$, where $(\Omega, \mathcal{F}, \mathbf{P})$ is a probability space, and $V : \Omega \to \mathcal{M}$ is a *measurable valuation* (that atributes for each sample-point $\omega \in \Omega$ a CPL valuation $V(\omega)$), where for all $\phi \in \text{Form}(\mathscr{L})$, we have that $V^{-1}(\phi) := \{\omega \in \Omega : V(\omega)(\phi) = 1\} \in \mathcal{F}$.

- the satisfaction relation $\vDash^p$ is recursively defined as:

  - $[r]_m := r$, for each $r \in \text{Alg}(\mathbb{R})$;
  - $[\int \phi]_m = \mathbf{P}\big(V^{-1}(\phi)\big) = \mathbf{P}\big(\{\omega \in \Omega : V(\omega)(\phi) = 1\}\big)$;
  - $[t_1 + t_2]_m := [t_1]_m + [t_2]_m$;
  - $[t_1 \times t_2]_m := [t_1]_m \times [t_2]_m$;

  - $m \vDash^p (t_1 \leq t_2)$ iff $[t_1]_m \leq [t_2]_m$;
  - $m \vDash^p (\neg \delta)$ iff $m \nvDash^p \delta$;
  - $m \vDash^p (\delta_1 \sqsupset \delta_2)$ iff ($m \nvDash^p \delta_1$ or $m \vDash^p \delta_2$),

  with $m \in \mathcal{M}^p$ and $\delta \in \text{Form}(\mathscr{L}^p)$.

We can see a lot of similarities with the probabilization made in Chapter 3. The only thing that is not clear to be equivalent to EPPL is how models are defined and how probabilities are determined.

In EPPL, each model was associated to $\mathbf{X} = (X_p)_{p \in \mathit{Var}(\mathcal{B})}$ a stochastic process over the probability space, where $X_p : \Omega \to \{0, 1\}$ is a Bernoulli random variable, and for each $\phi \in \text{Form}(\mathcal{B})$ induces a new Bernoulli r.v. X_ϕ recursively defined in an analogously way as CPL valuations.

In the CPL probabilization presented above, each model is associated to a measurable valuation V that for $\omega \in \Omega$ associates a CPL valuation $V(\omega)$. In order to obtain a semantical equivalent structure, we only need to have: $V(\omega)(p) = 1$ if and only if $X_p(\omega) = 1$, for all $p \in \mathit{Var}(\mathcal{B})$ (the equivalence to all non-atomic CPL formulas is guaranteed because recursively $X_\phi(\omega)$ and $V(\omega)(\phi)$ are defined in the same way).

Moreover, the probabilities are equaly defined: in EPPL, $[\int \phi]_m = \mathbf{P}(\{\omega \in \Omega : X_\phi(\omega) = 1\})$; and in the probabilization of CPL, $[\int \phi]_m = \mathbf{P}(\{\omega \in \Omega : V(\omega)(\phi) = 1\})$. The conditions are equivalent, and therefore $[\int \phi]_m$ is equal in both cases.

4.4.2 Probabilization of Modal Logic

In this section, we will give an idea of how to probabilize the Kripke modal logic ([HC96], [vB10]) using the probabilization operator defined in this section, and compare it with other existing probabilistic modal logics in literature (*e.g.* [SA07]). The syntax will be as usual, and to simplify, we will not consider the modal operator $\Diamond$ because it can be written as $(\sim\Box\sim)$.

$$\phi \;:=\; p \mid (\sim\phi) \mid (\phi \to \phi) \;\mid (\Box\phi)\;, \quad p \in Var(\mathscr{L})$$

Table 4.4: Modal Logic Syntax

Definition 4.4.2. A **Kripke Model** is a tuple $m = (W, R, f)$ where:

- W - set (called set of *worlds*);
- $R \subseteq W \times W$ - binary relation between worlds (called *acessibility relation*);
- $f : W \to \mathcal{P}\big(Var(\mathscr{L})\big)$ (called *labelling function*).

Example 4.4.3. Let $m = (W, R, L)$ be a Kripke model, where:
∘ $W = \{x_1, x_2, x_3, x_4, x_5\}$;
∘ $R = \{(x_1, x_2), (x_1, x_4), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_3, x_4), (x_5, x_2), (x_5, x_4), (x_5, x_5)\}$;
∘ $f(x_1) = \{p, q, r\}, f(x_2) = \{p\}, f(x_3) = \{p, r\}, f(x_4) = \{r\}, f(x_5) = \emptyset$.

Figure 4.1 representes graphically this Kripke model, where the arrow (x_i, x_j) is in the graph if and only if $(x_i, x_j) \in R$ (with $i, j \in \{1, 2, 3, 4, 5\}$).

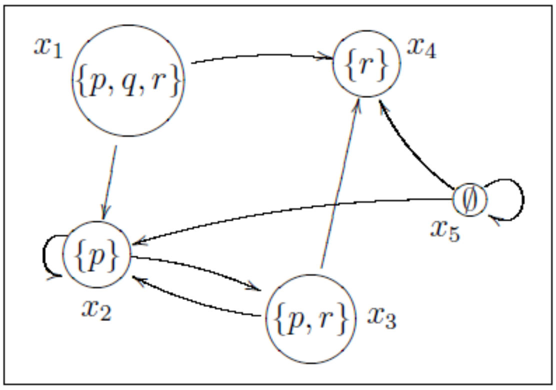

Figure 4.1: Graphic illustration of this Kripke model.

In this example, we can already see that putting probabilities in the arrows of this graph (that is, in the acessibility relation), we can define a probabilistic modal logic that in Probability Theory may correspond to Markov chains. Next, we will define the satisfaction relation to this logic as usual.

Definition 4.4.4. Let $m = (W, R, f)$ be a Kripke model and $x \in W$ a world. The satisfaction relation $\vDash_m$ for modal logic is recursively defined as follows:

- $x \vDash_m p$ iff $p \in f(x)$;
- $x \vDash_m (\sim\phi)$ iff $x \nvDash_m \phi$;
- $x \vDash_m (\phi_1 \to \phi_2)$ iff $x \nvDash_m \phi_1$ or $x \vDash_m \phi_2$;
- $x \vDash_m \Box\phi$ iff for all $y \in W$ if $(x, y) \in R$ then $y \vDash_m \phi$.

If $x \vDash_m \phi$, we say that the modal formula ϕ is satisfied in the world x (in the Kripke model m). Moreover, we say that m satisfies ϕ, denoted as $\vDash_m \phi$, if for each $x \in W$, we have that $x \vDash_m \phi$. And, we will denote as $\vDash \phi$ when ϕ is valid (that is, if $\vDash_m \phi$ for all models m).

Now, considering the probabilization operator defined before, we will apply it to this modal satisfaction system $\mathscr{L} = (L, \mathcal{M}, \vDash)$. The probabilization operator defined previoulsy transforms $\mathscr{L}$ in the satisfaction probabilistic system $\mathscr{L}^p = (L^p, \mathcal{M}^p, \vDash^p)$, defined by (here we will only mention the parts of the probabilization that change due to modal logic):

- $\mathcal{M}^p$ is the class of all models $m = (\Omega, \mathcal{F}, \mathbf{P}, V)$, where $(\Omega, \mathcal{F}, \mathbf{P})$ is a probability space, and $V : \Omega \to \mathcal{M}$ atributes for each sample-point $\omega \in \Omega$ a Kripke model $V(\omega)$, where for all modal formulas $\phi \in \text{Form}(\mathscr{L})$, we have that $V^{-1}(\phi) := \{\omega \in \Omega : \vDash_{V(\omega)} \phi\} \in \mathcal{F}$.
- $[\int \phi]_m = \mathbf{P}\big(V^{-1}(\phi)\big) = \mathbf{P}\big(\{\omega \in \Omega : \vDash_{V(\omega)} \phi\}\big) = \mathbf{P}\big(\{\omega \in \Omega : x \vDash_{V(\omega)} \phi, \text{ for all } x \in W\}\big)$

Translating this into words, with this probabilization of modal logic, the probabilities are associated with sets of Kripke models, *i.e.* to each sample-point is associated a modal Kripke model and the probability of a modal formula ϕ is calculated on the sets of Kripke models that satisfy this formula. Therefore, the probabilities are not associated with a fixed Kripke model (*e.g.* in its worlds or in it acessibility relation), and then this probabilization does not have the applicability that we were looking for.

Our interests with this probabilization would be given a Kripke model, we want to associate probabilities to something in this model. But to what would we associate the probability space? Our idea was to associate probabilities with the arrows (*i.e.* acessibility relation), but we must not associate one probability space to all the arrows because we want that for each world the sum of the probabilities associated to arrows that begins in this world is 1. Cleary, this probabilization does not do anything like that. In the last decades (since [Koz85]) probabilistic modal logics, like the one that we are looking for, have been a significant research topic.

Here we only try to probabilize modal logic (in general), but it would be interesting to study how to probabilize other logics, such as linear temporal logics ([Lic91]) and dynamic logics ([HKT00]).

Chapter 5

Conclusion and Further Work

Combining Logic with other areas of Mathematics is nowadays a very appealing topic of research. The main goal of this dissertation was to show that Probability Theory and Logic are two fields of knowledge not so far away from each other as we might think at first place.

Probabilistic logics join together two completely different areas of reasoning: Logic and Probability Theory. Chapter 2 serves as the basis for this work. It introduces the main topics necessary in order to understand the combination of logic with probabilities.

This dissertation is about the development of an Exogenous Probabilistic Logic, which was based on Classical Propositional Logic. In Chapter 3, we were able to prove Soundness and a weak version of Completeness of EPPL, and we actually proved that EPPL is not strongly complete (by given a counterexample). These properties (soundness and completeness) of a logic system plus its semantics presented are fundamental to make it an interesting system to explore.

Chapter 4 discusses some possible *evolutions* of EPPL. We introduced into EPPL semantics and Hilbert calculus other concepts that exist in other probabilistic logics and in Probability Theory. Namely, we introduced the concept of uncertainty that given a local formula corresponds to the probability of its negation. We also adapted a very important concept of Probability Theory: conditional (in this context, conditional between formulas).

We studied two ways of how to deal with conditional in EPPL. The first consisted in keeping the base logic unchanged, and introduce the conditional only as a probability (that is, as a real probabilistic term), and we defined two new axioms and proved that they are sound. A major challenge here will be to prove that EPPL with these conditionals is still weakly complete. Our idea is to modify the satisfaction (SAT) algorithm so that it incorporates these new terms, and adapt the proof of Theorem 3.5.6 for this new SAT algorithm. The second way was to introduce conditional as a local formula by changing CPL into a new logic with these conditional formulas - *Suppositional logic*. We defined a semantics and a deductive system for this logic, and we were actually able to prove its soundness. Again, a big challenge will be to prove that this suppositional logic is complete (at least, weakly complete).

We have also presented the idea of a language for a new probabilistic logic with these conditional formulas as local formulas, which we designated as EPPLC. Its semantics and Hilbert calculus will be similar to those of EPPL, but instead of considering the CPL definitions of valuation and tautology, we will consider now the concepts of u-valutions and u-valid formulas, which are extensions of CPL to a logic with three truth values ($\{0, \mathrm{u}, 1\}$). Our goal, in the future, will be to prove that at least for one of these approaches we have a complete logic (ideally, prove that both approaches that introduce conditionals in EPPL are complete).

In some works of Mateus, Baltazar and Nagarajan ([BMNP07], [BM09]) it is introduced a probabilistic linear temporal logic (EPLTL) that is obtained by enriching a probabilistic propositional logic with linear temporal modalities. An interesting approach is to include in this logic conditional formulas (as we did in EPPL), and try to understand how these temporal modalities behave with conditionals.

In this dissertation, we analyze several issues of probabilistic logics. The most challenging one was how to take an arbitrary logic system, and make it a probabilistic one. In Section 4.4, we defined an abstract way to do this for any satisfaction system. Future work could be to consider a first-order logic or a modal logic as our base logic, develop a probabilistic global logic (construct its semantics and Hilbert calculus) and study some applications.

In [RSS16], it is made a slightly different approach that consists of assigning probabilities not to CPL formulas, but to CPL valuations. In fact, it is referred that assigning probabilities to formulas or to valuations is equivalent (in terms of EPPL).

Theoretically, although Probabilistic logic can be applied to a large number of real problems, in practice, it is rarely so. The idea of this work was to develop a strong theoretical base, leaving open the practical applicability of our logic. In particular, we could try to develop an automatic prover for EPPL.

Another attractive idea is to use this EPPL in some mathematical software, *e.g.* PRISM. This is a programming language intended for symbolic-statistical modeling. PRISM is a logical program in which facts have a parameterized probability distribution so that the program can be seen as a parameterized statistical model. In [RS13] is an idea on how to put a probabilistic logic into PRISM, and it consists of doing the same thing we did to EPPL (and possibly, with conditionals).

In Section 3.5, we present a satisfaction algorithm to EPPL formulas in order to prove its weak completeness. At first we thought about trying to implement the algorithm, however this implementation is not as trivial as it may seem. The probabilistic satisfibiability (PSAT) problem (we can see it in detail in [AP01]) is another topic related to this logic, and it is one of the biggest problems nowadays of artificial intelligence, logic and computational complexity ([HJdA^{+}99], [FD04], [dBCF13]).

One issue that was not considered in this work, but that we intend to, in the future, was the *model checking* problem for our probabilistic logic, that is, how to develop an algorithm that given a model m, an assignment ρ and an EPPL formula δ, returns a boolean value corresponding to the value of $(m, \rho) \vDash \delta$. In the context of EPPL, Baltazar already studied this issue (see [BM09], [BMN10]), and he proved that his model checking algorithm takes $\mathcal{O}(|\delta|.|\Omega|)$ time to decide if (m, ρ) satisfies δ. EPPL model checking is studied in more detail in [Hen09].

Bibliography

[Ada98] Ernest W. Adams. *A Primer of Probability Logic.* Center for the Study of Language and Informations Publications - Stanford, California, 1998.

[AL75] E. W. Adams and H. Levine. On the Uncertainties Transmitted from Premisses to Conclusions in Deductive Inferences. *Synthese*, Vol.30 No.3:429–460, 1975.

[AP01] Kim Allan Andersen and Daniele Pretolani. Easy cases of probabilistic satisfiability. *Annals of Mathematics and Artificial Intelligence*, 33(1):69–91, 2001.

[AR02] Jorge Almeida and Hugo Ribeiro. *Introdução à Lógica.* Departamento de Matemática Pura, Universidade do Porto, 2002.

[ASB+95] Adnan Aziz, Vigyan Singhal, Felice Balarin, Robert Brayton, and Alberto Vincentelli. It Usually Works: The Temporal Logic of Stochastic Systems. *Computer Aided Verification*, pages 155–165, 1995.

[Bal10] Pedro Baltazar. *Probabilization of Logic Systems.* PhD thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, December 2010.

[BdA95] Andrea Bianco and Luca de Alfaro. Model checking of probabilistic and nondeterministic systems. *15th Conference on Foundations of Software Technology and Theoretical Computer Science*, 1026:499–513, 1995.

[BM09] Pedro Baltazar and Paulo Mateus. Temporalization of probabilistic propositional logic. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5407 LNCS:46–60, 2009.

[BMN10] Pedro Baltazar, Paulo Mateus, and Rajagopal Nagarajan. Fixpoint Logics for Reasoning about Probabilistic Systems. Preprint, SQIG - IT and IST - TU Lisbon, pages 1–19, 2010.

[BMNP07] Pedro Baltazar, Paulo Mateus, Rajagopal Nagarajan, and Nikolaos Papanikolaou. Exogenous Probabilistic Computation Tree Logic. *Electronic Notes in Theoretical Computer Science*, 190(3):95–110, 2007.

[BPR03] Saugata Basu, Richard Pollack, and Marie Roy. *Algorithms in Real Algebraic Geometry.* Algorithms and Computation in Mathematics 10. Springer Berlin Heidelberg, 2nd ed edition, 2003.

[BRS06] Pedro Baltazar, Jaime Ramos, and Cristina Sernadas. Probabilistic and quantum institutions revisited. Preprint, CLC, Department of Mathematics, IST, Lisbon, 2006.

[Car50] Rudolf Carnap. *Logical Foundations of Probability.* The University of Chicago Press, 1st edition, 1950.

[CCFMS07] R. Chadha, L. Cruz-Filipe, P. Mateus, and A. Sernadas. Reasoning about probabilistic sequential programs. *Theoretical Computer Science*, 379(1-2):142–165, 2007.

[CIN05] Rance Cleaveland, S. Purushothaman Iyer, and Murali Narasimha. Probabilistic temporal logics via the modal mu-calculus. *Theoretical Computer Science*, 342(2-3):316–350, 2005.

[CSS05] C. Caleiro, A. Sernadas, and C. Sernadas. Fibring logics: past, present and future. *We will show them! Essays in Honour of Dov Gabbay, Vol. I*, pages 363–388, 2005.

[dBCF13] Glauber de Bona, Fabio G. Cozman, and Marcelo Finger. Generalized Probabilistic Satisfiability. *2013 Brazilian Conference on Intelligent Systems*, pages 182–188, 2013.

[Fag90] R Fagin. A logic for reasoning about probabilities. *Information and Computation*, 87(1-2):78–128, 1990.

[FD04] Marcelo Finger and Glauber De Bona. Probabilistic Satisfiability: Logic-Based Algorithms and Phase Transition. Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Brazil, 2004.

[FH94] Alan M. Frisch and Peter Haddawy. Anytime deduction for probabilistic logic. *Artificial Intelligence*, 69(1):93 – 122, 1994.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[Hai86] Theodore Hailperin. *Boole's Logic and Probability.* Studies in Logic and the Foundations of Mathematics 85. Elsevier Science Ltd, 2 sub edition, 1986.

[Hai96] Theodore Hailperin. *Sentential Probability Logic.* Bethlehem: Lehigh University Press and London: Associated University Presses, 1996.

[Haj01] Alan Hajek. Probability, Logic, and Probability Logic. *The Blackwell Guide to Philosophical Logic*, pages 362–384, 2001.

[HC96] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[Hen09] David Henriques. Model Checking Probabilistic Systems. Master's thesis, Instituto Superior Técnico Lisboa, 2009.

[HJdA+99] Pierre Hansen, Brigitte Jaumard, Marcus Poggi de Aragão, Fabien Chauny, and Sylvain Perron. Probabilistic Satisability with Imprecise Probabilities. *International Journal of Approximate Reasoning*, 24(2):171–189, 1999.

[HKT00] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic logic.* Foundations of Computer Science, 2000.

[HRWW11] Rolf Haenni, Jan-Willem Romeijn, Gregory Wheeler, and Jon Williamson. *Probabilistic Logic and Probabilistic Networks.* The centre piece of Progicnet, Springer, 2010, 2011.

[IG10] Magdalena Ivanovska and Martin Giese. Probabilistic Logic with Conditional Independence. *The CODIO project*, 2010.

[Kol33] A. Kolmogorov. *Foundations of the theory of probability.* Chelsea Publishing Co., 1st edition edition, 1933.

[Koz85] Dexter Kozen. A probabilistic PDL. *Journal of Computer and System Sciences*, 30(2):162–178, 1985.

[Lic91] Orna Lichtenstein. *Decidability, Completeness, and Extensions of Linear Temporal Logic.* PhD thesis, Weizmann Institute of Science, 1991.

[MNBM16] Alexandre Madeira, Renato Neves, Luís S. Barbosa, and Manuel A. Martins. A method for rigorous design of reconfigurable systems. *Science of Computer Programming*, 2016.

[MNM16] Alexandre Madeira, Renato Neves, and Manuel A. Martins. An exercise on the generation of many-valued dynamic logics. *Journal of Logical and Algebraic Methods in Programming*, 2016.

[Mor11] Andreia Mordido. Dynamic probabilistic epistemic logic towards information security. Master's thesis, Instituto Superior Técnico Lisboa, 2011.

[MS04] P. Mateus and A. Sernadas. Exogenous Quantum Logic. CLC, Department of Mathematics, IST, pages 1–11, 2004.

[MSS05] P. Mateus, A. Sernadas, and C. Sernadas. Exogenous semantics approach to enriching logics. *Essays on the Foundations of Mathematics and Logic*, 1:165–194, 2005.

[Nil86] Nils J. Nilsson. Probabilistic Logic. *Artificial Intelligente*, 28:71–87, 1986.

[Nil93] Nils J. Nilsson. Probabilistic Logic (revisited). *Artificial Intelligente*, 59:39–42, 1993.

[PW08] Witold A. Pogorzelski and Piotr Wojtylak. *Completeness Theory for Propositional Logics.* Studies in universal logic. Birkhäuser, 1st edition, 2008.

[Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230 – 245, 1963.

[Ram26] Frank Plumpton Ramsey. *Truth and Probability.* University Archive for the History of Economic Thought, 1926.

[RS13] Fabrizio Riguzzi and Terrance Swift. Probabilistic Logic Programming Under the Distribution Semantics. Preprint submitted to *Elsevier*, December 2013.

[RSS13] J. Rasga, A. Sernadas, and C. Sernadas. Fibring as biporting subsumes asymmetric combinations. IST - UL and SQIG - IT, Lisbon, Portugal, 2013.

[RSS16] Joao Rasga, Cristina Sernadas, and Amilcar Sernadas. On probability and logic. IST - UL and SQIG - IT, Lisbon, Portugal, pages 1–43, 2016.

[SA07] Afsaneh Shirazi and Eyal Amir. Probabilistic modal logic. *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 22:489–495, 2007.

[She73] V. B. Shekhtman. Two-dimensional modal logic. *Mathematical Notes of the Academy of Sciences of the USSR*, 23:417–424, 1973.

[SSC05] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 2005.

[Sto08] Marielle Stoelinga. An introduction to probabilistic automata. Department of Cumputer Engineering, University of California at Santa Cruz, 2008.

[Tar56] A Tarski. *Logic, Semantics, Metamathematics. Papers from 1923 to 1938.* Oxford Univ. Press, first edition, 1956.

[vB10] Johan F. A. K. van Benthem. Modal Logic for Open Minds. *Chart*, 199:1–390, 2010.

[VPKS13] Evgenii E. Vityaev, Leonid I. Perlovsky, Boris Ya Kovalerchuk, and Stanislav O. Speransky. Probabilistic dynamic logic of cognition. *Biologically Inspired Cognitive Architectures*, 6:159–168, 2013.

[Woj84] Ryszard Wojcicki. *Lectures on Propositional Logic.* Ossolineum, The Publishing House of the Polish Academy of Sciences, 1984.