UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

A probabilistic logic over equations and domain restrictions

Andreia Filipa Torcato Mordido

Supervisor: Doctor Carlos Manuel Costa Lourenço Caleiro

Thesis approved in public session to obtain the PhD Degree in Information Security

Jury final classification: Pass with Distinction

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Luca Viganò
Doctor Marcelo Finger
Doctor Paulo Alexandre Carreira Mateus
Doctor Carlos Manuel Costa Lourenço Caleiro

2017

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

A probabilistic logic over equations and domain restrictions

Andreia Filipa Torcato Mordido

Supervisor: Doctor Carlos Manuel Costa Lourenço Caleiro

Thesis approved in public session to obtain the PhD Degree in Information Security

Jury final classification: Pass with Distinction

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Luca Viganò, Full Professor, Faculty of Natural & Mathematical Sciences, King's College London, UK

Doctor Marcelo Finger, Professor Titular do Instituto de Matemática e Estatística da Universidade de São Paulo, Brasil

Doctor Paulo Alexandre Carreira Mateus, Professor Associado (com Agregação) do Instituto Superior Técnico da Universidade de Lisboa

Doctor Carlos Manuel Costa Lourenço Caleiro, Professor Associado do Instituto Superior Técnico da Universidade de Lisboa

Funding Institutions

Fundação para a Ciência e Tecnologia (FCT)
Fundação Calouste Gulbenkian
Instituto de Telecomunicações

2017

Resumo

Nesta tese, propomo-nos apresentar uma lógica que permita formalizar o raciocínio subjacente à análise de protocolos criptográficos, nomeadamente no contexto de *offline guessing attacks*. A conjugação dos parâmetros que caracterizam a análise de protocolos de segurança requere que a lógica seja dotada de três componentes fundamentais: equações, probabilidades e quantificadores.

Começamos por apresentar uma lógica (EqCL) que nos permite raciocinar sobre restrições equacionais e que resulta da combinação da lógica proposicional clássica, da lógica equacional e de quantificadores. Apresentamos uma axiomatização correta e completa para esta lógica, parametrizada por uma especificação equacional da base algébrica. No sentido de automatizar o raciocínio dedutivo, exploramos o problema de satisfatibilidade da lógica EqCL e apresentamos uma redução polinomial a SAT, sob a hipótese de que a teoria equacional subjacente é convergente. Inspirados pela análise do problema da satisfatibilidade para EqCL, propomo-nos explorar o problema da satisfatibilidade probabilística e estendê-lo a combinações lineares de fórmulas probabilísticas envolvendo fórmulas proposicionais. Definimos assim o problema GenPSAT e apresentamos uma redução polinomial a Programação Linear Inteira Mista. Uma vez implementada uma ferramenta que decide GenPSAT, estudamos o comportamento de transição de fase deste problema NP-completo. Estamos então em condições de definir a lógica pretendida: DEqPrL é a lógica probabilística definida sobre uma base algébrica que engloba equações e restrições de domínio. Apresentamos um sistema dedutivo correto e fracamente completo para DEqPrL parametrizado por uma especificação equacional da base algébrica e pelas respectivas restrições de domínio. Com base nos resultados obtidos para GenPSAT, apresentamos uma redução polinomial do problema de satisfatibilidade para DEqPrL a Satisfatibilidade Módulo Teorias, assumindo que a teoria equacional subjacente é convergente e que as restrições de domínio satisfazem uma propriedade adequada.

Ilustramos a aplicabilidade das lógicas apresentadas em vários exemplos, nomeadamente no contexto da análise de *offline guessing attacks* a protocolos criptográficos.

Palavras-chave: Lógica Probabilística, Lógica Equacional, Completude, Satisfatibilidade, *Offline Guessing Attacks*.

Abstract

In this thesis, we aim to provide a logic to deal with the reasoning required for the analysis of cryptographic protocols, namely in the context of offline guessing attacks. The envisaged logic should be able to cope with equations, probabilities and quantifiers in order to address the common features of cryptographic protocols analysis.

We start by presenting a logic (EqCL) able to state and reason about equational constraints, by combining aspects of classical propositional logic, equational logic and quantifiers. We provide a sound and complete axiomatization parametrized by an equational specification of the algebraic basis. Then, we explore the satisfiability problem for EqCL and present a polynomial reduction to SAT, under the assumption that the underlying equational theory is convergent. Inspired by the satisfiability results for EqCL, we aim at extending the scope of the probabilistic satisfiability problem by allowing linear combinations of probabilistic assignments of values to propositional formulas. We define the GenPSAT problem and present a polynomial reduction of GenPSAT to Mixed-Integer Programming. With a solver in hands, we study the phase transition behaviour of this NP-complete problem. Once collected all the necessary ingredients, we present DEqPrL - the probabilistic logic over an algebraic basis that enables one to reason about equations and domain restrictions. We provide a sound and weak complete deductive system for DEqPrL, parametrized by an equational specification of the algebraic basis coupled with the intended domain restrictions. Driven by the developments with GenPSAT, we provide a polynomial reduction of the satisfiability problem for DEqPrL to the Satisfiability Modulo Theories, assuming that the underlying equational theory is convergent and that the axiomatization of domain restrictions enjoys a suitable subterm property.

Some relevant examples that illustrate the usefulness of the logics, namely regarding the static analysis of offline guessing attacks to cryptographic protocols, are also presented.

Keywords: Probabilistic Logic, Equational Logic, Completeness, Satisfiability, Offline Guessing Attacks.

Acknowledgments

First of all, I would like to express my most sincere gratitude to my supervisor, Professor Carlos Caleiro, for the guideness and persistent support, for his patience and for the many crucial advices over these years. I am truly grateful!

My deepest gratitude to my family for all the love, encouragement and the trust they have always placed in me. To my brother, Gonçalo, I would also like to thank the long philosophical conversations and his ability to take care of me by giving the most candid advices ever. To my mother, Cristina, I offer my heartfelt thanks for always keeping me safe when the world seems to turn dark. To my father, António, I thank all the concern and support.

To my co-author Filipe Casal I would like to thank all the fruitful discussions, his willingness, and the long working hours, even when there was nobody left in the department. I also wish to thank Iolanda and Guilherme all the jokes and laughs. Without the three of them it would not have had the same meaning.

I would like to thank everyone at the Security and Quantum Information Group at IT for their support. To "our" meta-supervisor, Professor Amílcar Sernadas, a word of thanks for the recommendations throughout this long journey.

Finally, I thank the funding sources that made this work possible:

- Fundação para a Ciência e Tecnologia (FCT) through the PhD scholarship with reference SFRH/BD/77648/2011;
- Fundação Calouste Gulbenkian through the award in Programa de Estímulo à Investigação 2011;
- Instituto de Telecomunicações through the research grants BIM/n.º 100 and IT-EEA/50008-GenPSAT;
- Project ComFormCrypt from Instituto de Telecomunicações for enabling me to attend to several interesting conferences.

Contents

List of Algorithms

Introduction

Information security is a long-standing concern given its military, political, social, and economic implications. In this era of global electronic communication and commerce, it is no wonder that the subject deserves increasing attention. It is an interdisciplinary area that, besides all the engineering aspects, requires a deep understanding of algebra and number theory, information theory, computational complexity, probabilities, concurrency and distribution (see [104]).

Modern cryptography was driven by Diffie and Hellman's proposal in 1976 [52]; Diffie and Hellman proposed an encryption notion contrasting with the traditional symmetric encryption scheme - the asymmetric cryptography. Asymmetric cryptography came to establish a new paradigm of secure communication and thereafter it has been widely used to ensure the security of communications. Nevertheless, the well-known attack to the Needham-Schroeder public-key protocol [74], only discovered 18 years after its disclosure [86], highlighted the need for using formal methods in the analysis of cryptographic protocols.

The analysis of secure communication protocols relies essentially in two approaches: a *computational approach* in which the computational complexity issues, resource-bounded attackers and probabilities of attacks are taken into consideration [21, 31]; and the *formal approach*, based on the idealistic hypothesis of perfect cryptography [4, 27], that adopts the Dolev-Yao attacker [53]. The former approach, for being much closer to reality, has had a significant success but is often far from being scalable or amenable to automation. On the contrary, the latter departs from an idealization of the cryptographic primitives, but has been very successful in practice while also being largely automatable and, in many cases, scalable. The formal approach is meaningful because, beyond the specific security of the cryptographic primitives being used, all the communications take place in a hostile open network in which any malicious agent can interfere. In fact, most of the formal approaches are primarily directed towards finding attacks on protocols, rather than on proving their correctness, which is actually undecidable in general due to the fact that the search space is infinite. Their underlying computational and analytic models include higher-order logic theories, rewriting and narrowing in equational theories, process calculi, temporal and epistemic logics, and strand spaces [1,4,8,9,11,27,54,57,74,75,79,93].

The advantages that arise from the combination of both methodologies are widely recognized (e.g. by the use of formal methods for predicting the impact of small implementation details, as we will see in Example 4.5.3) and there is a great effort to reduce the gap between the two approaches [5–7,14,16,25,47,70,83,94], using more sophisticated probabilistic models and taking into account computational complexity issues [70].

Naturally, 'formal methods' should be read as 'logics', but in reality things are not that straightforward. In fact, the problem at hand is usually so intricate that suitable fully-fledged logics have not yet been developed, and the reasoning is usually carried over in an underspecified higher-order metalogic, often incorporating many ingredients: ranging from equational to probabilistic reasoning, from communication and distribution, to temporal or epistemic reasoning [44].

Such logics would enable the formalization of the very important fact that security is not an absolute notion. On the contrary, the realistic view of security should be relative - a security property should only be considered to hold as long as the probability of an attack assuming a computationally bounded attacker (typically with polynomially-bounded computational power) is negligible. The proper integration of such logical notions into the formal approach to security analysis has a very positive impact on understanding and shortening the gap towards the computational approach.

The research on this thesis seeks to contribute to the reduction of the gap between both approaches and aims at developing a logic able to deal with the reasoning required for the analysis of cryptographic protocols. The main purpose is mostly to provide a formal ground for the reasoning of a Dolev-Yao attacker breaking a cryptographic protocol, namely in the context of the static analysis of *offline guessing attacks* [18]. In *offline guessing*, typically, an attacker eavesdrops the network and gets hold of a number of messages exchanged by the parties. These messages are usually generated from random data and ciphered using secret passwords or weak keys, being immediately unreadable, but often are known to have strong algebraic relationships between them. These algebraic relationships are captured into the logic by means of equations between terms whose underlying set of generators represents the random data, whereas the cryptographic primitives are represented through function symbols.

If the attacker tries to guess the secret keys (a realistic hypothesis in many scenarios, including human-picked passwords, or protocols involving devices with limited computational power), he may use these relationships to validate his guess. The use of quantifiers enables us to handle the many possible scenarios that emerge from attacker's guesses. When the guesses and the random data arise from a probability distribution, we can take a step forward and quantify our analysis by estimating the success of attacks. The analysis of the attacker's reasoning is enabled by incorporating classical reasoning into the logic.

We can further take a bold step towards cryptanalysis and consider a setting where the usual Dolev-Yao intruder is extended with some cryptanalytic power [41, 83]. In this setting, besides their algebraic relationships, the random data, the guesses and the messages are known to additionally comply with certain domain restrictions that may be crucial to the attacker analysis.

Against this backdrop, we aim at developing a logic that combines aspects from classical logic and equational logic with an exogenous-like approach [77] to quantitative probabilistic reasoning. The non-trivial way in which all these components interact, reveals a long (and very interesting!) journey ahead of us.

The structure of this thesis can be seen has a peculiar semilattice whose nodes consist of the 3 main components of this logic - classical reasoning, equational reasoning and probabilities - and the *supremum* of the 3 nodes is the probabilistic logic over an algebraic basis, that we aim to achieve.

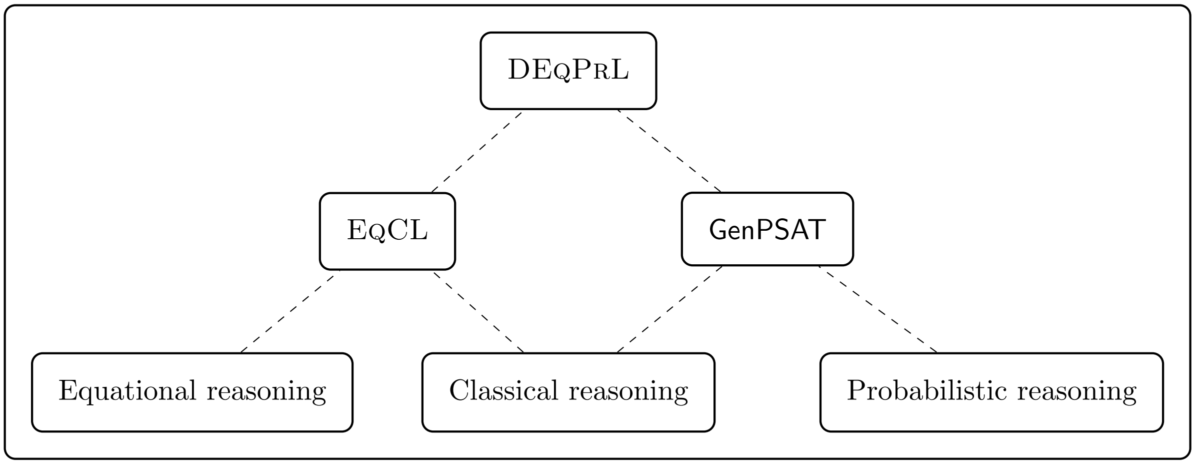


Figure 1: Structure of the dissertation.

Along the dissertation, we will be moving from the bottom to the top of Figure 1. We start with an overview of the logics that characterize each of the components of the logic, namely: propositional classical logic [34, 81], equational logic [24, 80, 105], and probabilistic logic [58]. Then, we ascend one step higher and present a logic with a classical structure over an equational basis (EqCL). To ensure that we are prepared to move to the top, we develop a solver for a generalization of the probabilistic satisfiability problem (see [59,60]), the GenPSAT problem. Finally, we present the probabilistic logic over an algebraic basis (DEqPrL) that we envisaged. Due to the main motivation of this work relying in information security concepts, we will make sure to bring some relevant examples throughout the text.

Outline

The dissertation is organized in 5 chapters organized as follows:

Preliminaries: The preliminary chapter provides the ground for the upcoming chapters and consists of a survey of the logics for the most basic components underlying our work, namely, classical propositional logic, probabilistic logic and equational logic. We take the opportunity to target our algebraic approach for the motivation that drove this work: information security concepts. A great part of the contents of the first chapter may be skipped for those who are already familiar with the referred logics. However, we introduce some original concepts at the end of this preliminary chapter, when we address some extensions of equational logic.

Equation-Based Classical Logic (EqCL): In Chapter 2, we propose and study a logic able to state and reason about equational constraints, by combining aspects of classical propositional logic, equational logic, and quantifiers. The logic has a classical structure over an algebraic basis, and a form of universal quantification distinguishing between local and global validity of equational constraints. We present a sound and complete axiomatization for the logic, parameterized by an equational specification of the algebraic basis. We provide a polynomial reduction of the satisfiability problem for the logic to the SAT problem and show that the logic is decidable under the assumption that its algebraic basis is given by a convergent rewriting system, thus covering an interesting range of examples. As an application, we analyze offline guessing attacks to security protocols, where the equational basis specifies the algebraic properties of the cryptographic primitives. This work was developed with Carlos Caleiro and published in [84].

Generalized Probabilistic Satisfiability (GenPSAT): In Chapter 3, we study a generalized probabilistic satisfiability problem - GenPSAT - which consists in deciding the satisfiability of linear inequalities involving probabilities of classical propositional formulas. GenPSAT is proved to be NP-complete and we present a polynomial reduction to Mixed-Integer Programming. Capitalizing on this translation, we implement and test a solver for the GenPSAT problem. As previously observed for many other NP-complete problems, we are able to detect a phase transition behaviour for GenPSAT. This work was developed with Carlos Caleiro and Filipe Casal, was presented in the Workshop on Logical and Semantic Frameworks with Applications LSFA 2016, and was submitted for publication, see [29].

Probabilistic Logic over Equations and Domain Restrictions (DEqPrL): In Chapter 4, we propose and study a probabilistic logic over an algebraic basis, including equations and domain restrictions. The logic combines aspects from classical logic and equational logic with an exogenous- -like approach to quantitative probabilistic reasoning. We present a sound and (weakly) complete axiomatization for the logic, parameterized by an equational specification of the algebraic basis coupled with the intended domain restrictions. We show that the satisfiability problem for the logic is decidable, under the assumption that its algebraic basis is given by means of a convergent rewriting system, and, additionally, that the axiomatization of domain restrictions enjoys a suitable subterm property. For this purpose, we provide a polynomial reduction to Satisfiability Modulo Theories with respect to the theory of quantifier-free linear arithmetic over the integers and reals (QF_LIRA). As a consequence, we get that validity in the logic is also decidable. Furthermore, under the assumption that the rewriting system that defines the equational basis underlying the logic is also subterm convergent, we show that the resulting satisfiability problem is NP-complete, and thus the validity problem is coNP-complete. We also provide an implementation of a solver for the satisfiability problem and test the logic with meaningful information security examples, proving that it can handle some implementation details of cryptographic protocols formally and then conclude how do they compromise security, by estimating the probability of attacks. This work was developed with Carlos Caleiro and submitted for publication, see [85].

Conclusions and Future Work: In Chapter 5, we briefly summarize the main achievements of this thesis, assess our contributions and discuss some limitations, room for improvement and future research.

Summary of Contributions

Given the context of this dissertation, we summarize those that we consider to be the main contributions of this thesis:

- the development of a logic (EqCL) combining aspects of classical propositional logic, equational logic and quantifiers in order to be able to state and reason about equational constraints in both local and global contexts;

- the polynomial reduction of the satisfiability problem for EqCL to SAT under the assumption that the equational basis is given by means of a convergent rewriting system;

- the decidability result for EqCL, proved using the satisfiability algorithm that is based on SAT;

- the polynomial reduction of a generalization of the probabilistic satisfiability problem (GenPSAT) with linear inequalities involving probabilities of classical propositional formulas to Mixed-Integer Programming;

- the implementation of a GenPSAT solver and subsequent analysis of the phase transition behaviour;

- the development of a logic (DEqPrL) combining aspects from classical propositional logic, equational logic with domain restrictions, probabilities and quantifiers, by providing a sound and (weakly) complete deductive system parametrized by an equational specification of the algebraic basis coupled with the intended domain restrictions;

- the decidability procedure for the satisfiability problem for DEqPrL by reduction to QF_LIRA, under the assumption that the equational basis is given by means of a convergent rewriting system and that the axiomatization of the domain restrictions enjoy a suitable subterm property;

- the decidability result for DEqPrL, proved using the satisfiability algorithm that, in its turn, is based on QF_LIRA;

- the application of the logic to meaningful examples in information security, namely by verifying and estimating the sucess of offline guessing attacks to cryptographic protocols by a Dolev-Yao intruder with some cryptanalytic power.

Chapter 1

Preliminaries

In this chapter, we survey non-original preliminary concepts and results necessary to understand the work presented in this thesis. This overview is not exhaustive and we often mention references in the literature for a more in depth treatment of these matters. Nonetheless, it enables us to fix some notation.

The chapter is outlined as follows: in Section 1.1 we introduce some brief and well-known notions of logic; in Section 1.2 we overview the probabilistic logic proposed by Fagin, Halpern and Megiddo in [58]; Section 1.3 consists of a brief survey of equational logic.

1.1 Logic

As the main topic of our study, *logic* deserves a careful treatment and an uniform approach throughout the whole manuscript. In this sense, we do not aim to introduce original content in this section, but only to survey some well-known topics, definitions and results that can easily be found in any textbook on logic [102,106,110] and universal algebra [24,80].

We follow the lines of [106] and look at logical consequence as the central aspect of logic. The logical consequence gives the most fundamental meaning to the language of the logic. For this purpose, we consider both model-theoretic and proof-theoretic frameworks providing two distinct (but often interrelated) approaches for consequence relation: the semantic entailment and the deducibility relation.

1.1.1 Language

A formula in the language is a syntactic object, to which it can be assigned an interpretation. Formulas are composed by a sequence of atoms and logical connectives, that usually follow some rules imposed by a given grammar.

Definition 1.1.1. Let A be a set of *atoms* and C be a finite set of *logical connectives*, to which is associated an assignment of non-negative integers representing its *arity* (number of arguments), $\mathsf{ar} : \mathsf{C} \to \mathbb{N}$. The *language* L over A with connectives C is defined by the following grammar:

$$\mathrm{L} ::= \mathsf{A} \mid \mathsf{C}\,(\mathrm{L}, \dots, \mathrm{L}).$$

The inductive feature of the languages turns out to be very useful in our proofs, since we often explore the inductive structure of the formulas.

The notion of subformula arises very naturally.

Definition 1.1.2. Let A be a set of atoms, C be a finite set of logical connectives with the corresponding arities given by $\mathsf{ar} : \mathsf{C} \to \mathbb{N}$ and L be a language over A with connectives C. We define the set of *subformulas* of $\varphi \in \mathrm{L}$ inductively by:

- $\mathsf{subform}(\alpha) = \{\alpha\}$;
- $\mathsf{subform}(c(\varphi_1, \dots, \varphi_{\mathsf{ar}(c)})) = \{c(\varphi_1, \dots, \varphi_{\mathsf{ar}(c)})\} \cup \bigcup_{i=1}^{\mathsf{ar}(c)} \mathsf{subform}(\varphi_i)$,

where $\alpha \in \mathsf{A}$, $c \in \mathsf{C}$ and $\varphi_1, \dots, \varphi_{\mathsf{ar}(c)} \in \mathrm{L}$. Given a set of formulas $\Phi \subseteq \mathrm{L}$, we denote by $\mathsf{subform}(\Phi)$ the set of subformulas of all formulas in Φ, $\mathsf{subform}(\Phi) = \bigcup_{\varphi \in \Phi} \mathsf{subform}(\varphi)$.

Definition 1.1.3. Let A be a set of atoms, C be a finite set of logical connectives with the corresponding arities given by $\mathsf{ar} : \mathsf{C} \to \mathbb{N}$ and L be a language over A with connectives C. We define the *length of a formula* $\varphi \in \mathrm{L}$, which we denote by $|\varphi|$, inductively as follows:

- $|\alpha| = 1$;
- $|c(\varphi_1, \dots, \varphi_{\mathsf{ar}(c)})| = 1 + |\varphi_1| + \dots + |\varphi_{\mathsf{ar}(c)}|$,

where $\alpha \in \mathsf{A}$, $c \in \mathsf{C}$ and $\varphi_1, \dots, \varphi_{\mathsf{ar}(c)} \in \mathrm{L}$.

1.1.2 Consequence relation

In order to establish the fundamental concept of consequence relation, let us fix some notation: given a set L, let us denote by $\wp(\mathrm{L})$ the set of all subsets of L.

Definition 1.1.4. Let L be a set. A *consequence relation* on L is a relation $\vdash \subseteq (\wp(\mathrm{L}) \times \mathrm{L})$ such that the following conditions hold, for every $\varphi \in \mathrm{L}$ and $\Phi, \Psi \subseteq \mathrm{L}$:

- if $\varphi \in \Phi$, then $\Phi \vdash \varphi$;
- if $\Phi \vdash \varphi$ and $\Phi \subseteq \Psi$, then $\Psi \vdash \varphi$;
- if $\Phi \vdash \varphi$ and for every $\psi \in \Phi$, $\Psi \vdash \psi$, then $\Psi \vdash \varphi$.

The consequence relation $\vdash$ is *finitary* if it additionally satisfies:

- if $\Phi \vdash \varphi$, then there exists a finite set Ψ such that $\Psi \subseteq \Phi$ and $\Psi \vdash \varphi$.

Definition 1.1.5. A *logic* is a pair $\mathcal{L} = \langle \mathrm{L}, \vdash \rangle$ where:

- L is a set of *formulas*;

- $\vdash \subseteq (\wp(\mathrm{L}) \times \mathrm{L})$ is a consequence relation.

Definition 1.1.6. Let $\mathcal{L} = \langle \mathrm{L}, \vdash \rangle$ be a logic. Given $\varphi \in \mathrm{L}$ and $\Phi \subseteq \mathrm{L}$ we say that φ is a *consequence* of Φ in $\mathcal{L}$ when $\Phi \vdash \varphi$. If $\Phi = \varnothing$, φ is said to be a *theorem* in $\mathcal{L}$ and we simply write $\vdash \varphi$. A set $\Phi \subseteq \mathrm{L}$ is *inconsistent* in $\mathcal{L}$ if $\Phi \vdash \varphi$ for every $\varphi \in \mathrm{L}$, otherwise Φ is said to be *consistent* in $\mathcal{L}$. Furthermore, a formula $\varphi \in \mathrm{L}$ is said to be consistent in $\mathcal{L}$ provided that the set $\{\varphi\}$ is consistent.

When the logic $\mathcal{L}$ is clear from context, we often omit the reference to $\mathcal{L}$.

Definition 1.1.7. Let $\mathcal{L} = \langle \mathrm{L}, \vdash \rangle$ be a logic. $\Psi \subseteq \mathrm{L}$ is said to be a *maximal consistent set* in $\mathcal{L}$ if:

- Ψ is consistent in $\mathcal{L}$;

- for every $\Phi \subseteq \mathrm{L}$ such that $\Psi \subset \Phi$, Φ is inconsistent in $\mathcal{L}$.

Maximal consistent sets are often called *complete theories* (see [98, 106]). They will be very useful provided the following existential result that is originally attributed to Adolf Lindenbaum and was published by Alfred Tarski in [106].

Lemma 1.1.8 (Lindenbaum's Lemma)**.** *Let $\mathcal{L} = \langle \mathrm{L}, \vdash \rangle$ be a logic where $\vdash$ is a finitary consequence relation on* L*. If $\Phi \subseteq \mathrm{L}$ is consistent in $\mathcal{L}$, then there exists a maximal consistent set Ψ in $\mathcal{L}$ that contains Φ, $\Phi \subseteq \Psi$.*

Often we are faced with two main approaches for consequence relations on the same language: the semantic (model-theoretic) consequence relation and the syntactic (proof-theoretic) consequence relation. The former one stands for ensuring that every model satisfying premises also satisfies the conclusions, whereas the second one stands for the derivation of conclusions from premises with the applications of rules of inference.

Semantic consequence relation

When the logical validity is sustained in the absence of a counterexample, we are in the presence of a model-theoretic approach for consequence relation. In this semantic approach, an argument is valid if in any model where the premises hold, the conclusion also holds.

Definition 1.1.9. A *satisfaction system* is a triple $\mathfrak{S} = \langle \mathrm{L}, \mathcal{M}, \Vdash \rangle$ composed by:

- a set of *formulas* L;
- a class of *models* $\mathcal{M}$;
- a *satisfaction relation* $\Vdash \subseteq (\mathcal{M} \times \mathrm{L})$.

Given a formula $\varphi \in \mathrm{L}$ and a model $M \in \mathcal{M}$, we say that M *satisfies* φ if $M \Vdash \varphi$. A formula φ is *satisfiable* in $\mathfrak{S}$ if there exists a model $M \in \mathcal{M}$ that satisfies φ; φ is *valid* (in $\mathfrak{S}$) if for every $M \in \mathcal{M}$, $M \Vdash \varphi$.

We extend the notion of satisfaction by requiring that a model $M \in \mathcal{M}$ satisfies a set of formulas $\Phi \subseteq \mathrm{L}$ (in symbols, $M \Vdash \Phi$) if $M \Vdash \varphi$ for each $\varphi \in \Phi$.

Definition 1.1.10. Let $\mathfrak{S} = \langle \mathrm{L}, \mathcal{M}, \Vdash \rangle$ be a satisfaction system. The formulas $\varphi_1, \varphi_2 \in \mathrm{L}$ are *equivalent* provided that for every model $M \in \mathcal{M}$, $M \Vdash \varphi_1$ if and only if $M \Vdash \varphi_2$.

Definition 1.1.11. Consider two satisfaction systems $\mathfrak{S}_1 = \langle \mathrm{L}_1, \mathcal{M}_1, \Vdash_1 \rangle$ and $\mathfrak{S}_2 = \langle \mathrm{L}_2, \mathcal{M}_2, \Vdash_2 \rangle$. The formulas $\varphi_1 \in \mathrm{L}_1$ and $\varphi_2 \in \mathrm{L}_2$ are *equisatisfiable* provided that φ_1 is satisfiable in $\mathfrak{S}_1$ if and only if φ_2 is satisfiable in $\mathfrak{S}_2$.

Definition 1.1.12. Let $\mathfrak{S} = \langle \mathrm{L}, \mathcal{M}, \Vdash \rangle$ be a satisfaction system. We define the *semantic entailment* $\vDash_{\mathfrak{S}} \subseteq (\wp(\mathrm{L}) \times \mathrm{L})$ as follows: $\Phi \vDash_{\mathfrak{S}} \varphi$ provided that for every model $M \in \mathcal{M}$, $M \Vdash \varphi$ whenever $M \Vdash \Phi$. A formula $\varphi \in \mathrm{L}$ is said to be *entailed* in $\mathfrak{S}$ by $\Phi \subseteq \mathrm{L}$ if $\Phi \vDash_{\mathfrak{S}} \varphi$.

A semantic entailment $\vDash_{\mathfrak{S}}$ is said to be *compact* if $\Phi \vDash_{\mathfrak{S}} \varphi$ implies that there exists a finite set $\Psi \in \wp(\mathrm{L})$ with $\Psi \subseteq \Phi$ such that $\Psi \vDash_{\mathfrak{S}} \varphi$.

Proposition 1.1.13. *Let $\mathfrak{S} = \langle \mathrm{L}, \mathcal{M}, \Vdash \rangle$ be a satisfaction system. The semantic entailment $\vDash_{\mathfrak{S}}$ is a consequence relation.*

Proof. To prove the first condition required for being a consequence relation, let $\Phi \subseteq \mathrm{L}$, $M \in \mathcal{M}$ and assume that the $M \Vdash \Phi$. We can immediately conclude that $M \Vdash \varphi$ for every $\varphi \in \Phi$, hence $\Phi \vDash_{\mathfrak{S}} \varphi$.

Now let us consider two sets of formulas Φ, Ψ such that $\Phi \subseteq \Psi \subseteq \mathrm{L}$ and a formula $\varphi \in \mathrm{L}$. Assume that $\Phi \vDash_{\mathfrak{S}} \varphi$. Then, let $M \in \mathcal{M}$ be any model such that $M \Vdash \Psi$. Since $\Phi \subseteq \Psi$, this means that $M \Vdash \Phi$, so $M \Vdash \varphi$. We conclude that $\Psi \vDash_{\mathfrak{S}} \varphi$.

Finally, let $\Phi, \Psi \subseteq \mathrm{L}$ be any two sets of formulas and $\varphi \in \mathrm{L}$ be a formula. Assume that $\Phi \vDash_{\mathfrak{S}} \varphi$ and for every $\psi \in \Phi$, $\Psi \vDash_{\mathfrak{S}} \psi$. Then, let $M \in \mathcal{M}$ be any model satisfying Ψ. It means that $M \Vdash \psi$ for every $\psi \in \Phi$, therefore $M \Vdash \Phi$. So, $\Psi \vDash_{\mathfrak{S}} \varphi$. □

Sometimes we denote the semantic entailment $\vDash_{\mathfrak{S}}$ by $\vDash_{\mathcal{M}}$, for a satisfaction system $\mathfrak{S} = \langle \mathrm{L}, \mathcal{M}, \Vdash \rangle$.

Syntactic consequence relation

Next, we focus on a proof-theoretic approach for logical consequence, where the validity of an argument is based on the existence of a proof of the conclusion from the premises. There are many proof-calculi in the literature [17,63,102,109,110] and some of them could even be more suitable and informative for the proofs that we intend to do, however we focus on probably the most common proof-theoretic approach to define consequence relations: (Hilbert-style) deductive systems.

Definition 1.1.14. Let L be a set of formulas. An *inference rule* over L is a pair $\langle \mathrm{Prem}, \mathrm{Conc} \rangle$ where:

- $\mathrm{Prem} \subseteq \mathrm{L}$ is a set of *premises*,
- $\mathrm{Conc} \in \mathrm{L}$ is a *conclusion.*

An inference rule is said to be *finitary* if $\mathrm{Prem} \in \wp(\mathrm{L})$ is a finite set. We denote by $\mathrm{InfR}(\mathrm{L})$ and $\mathrm{InfR}_{\mathrm{fin}}(\mathrm{L})$ the set of inference rules and the set of finitary inference rules over L, respectively.

Given $\varphi, \varphi_1, \ldots, \varphi_n, \ldots \in \mathrm{L}$, we denote an inference rule $\langle \{\varphi_1, \ldots, \varphi_n, \ldots\}, \varphi \rangle$ by $\frac{\varphi_1 \cdots \varphi_n \cdots}{\varphi}$. An *axiom* stands for an inference rule without premises. We denote an axiom $\langle \varnothing, \varphi \rangle$ simply by φ.

Definition 1.1.15. Let L be a set of formulas. A *deductive system* is a set of finitary inference rules $\mathcal{H} \subseteq \mathrm{InfR}_{\mathrm{fin}}(\mathrm{L})$.

Definition 1.1.16. Let L be a set of formulas and $\mathcal{H}$ be a deductive system. A *deduction* in $\mathcal{H}$ from a set of formulas $\Phi \subseteq \mathrm{L}$ is a finite sequence of formulas D such that for each $\psi \in D$ at least one of the following conditions holds:

- $\psi \in \Phi$;
- ψ is an axiom[1] of $\mathcal{H}$;
- ψ is a conclusion of an inference rule[1] of $\mathcal{H}$ whose premises occur earlier in D.

Let $\varphi \in \mathrm{L}$ and $\Phi \subseteq \mathrm{L}$. A *deduction of* φ *from* Φ in $\mathcal{H}$ is a deduction D from Φ in which φ is the last formula occurring in D. The formula φ is said to be *deducible from* Φ *in* $\mathcal{H}$, and we write $\Phi \vdash_{\mathcal{H}} \varphi$, if there exists a deduction of φ from Φ. The relation $\vdash_{\mathcal{H}} \subseteq (\wp(\mathrm{L}) \times \mathrm{L})$ is called a *deducibility relation.*

[1] When the logic has an implicit notion of substitution, we are often faced with an axiom schemata instead of axioms and inference rules. In this case, we need to consider all the instances of each axiom schema and all the instances of each rule schema.

Proposition 1.1.17. *Let* L *be a set of formulas and* $\mathcal{H}$ *be a deductive system. The deducibility relation* $\vdash_{\mathcal{H}}$ *is a finitary consequence relation.*

Proof. Let $\Phi \subseteq \mathrm{L}$ and notice that we can deduce every $\varphi \in \Phi$ from Φ, therefore the first condition for being a consequence relation holds, $\Phi \vdash_{\mathcal{H}} \varphi$.

Then, let us consider two sets of formulas Φ, Ψ such that $\Phi \subseteq \Psi \subseteq \mathrm{L}$ and a formula $\varphi \in \mathrm{L}$. Assume that $\Phi \vdash_{\mathcal{H}} \varphi$ and notice that, since $\Phi \subseteq \Psi$, the same deduction allows us to conclude $\Psi \vdash_{\mathcal{H}} \varphi$.

For the third condition, let $\Phi, \Psi \subseteq \mathrm{L}$ be any two sets of formulas and $\varphi \in \mathrm{L}$ be a formula. Assume that $\Phi \vdash_{\mathcal{H}} \varphi$ and $\Psi \vdash_{\mathcal{H}} \psi$, for every $\psi \in \Phi$. To conclude that $\Psi \vdash_{\mathcal{H}} \varphi$, consider the deduction D that led to $\Phi \vdash_{\mathcal{H}} \varphi$ and substitute each occurrence of a formula $\psi \in \Phi$ in D by its deduction from Ψ.

To check finitaryness, let $\Phi \subseteq \mathrm{L}$, $\varphi \in \mathrm{L}$ and assume that $\Phi \vdash_{\mathcal{H}} \varphi$. Since the deduction system $\mathcal{H}$ only contains finitary inference rules and the deduction of φ from Φ is finite, it follows that there exists a finite set $\Psi \in \wp(\mathrm{L})$ such that $\Psi \subseteq \Phi$ and $\Psi \vdash_{\mathcal{H}} \varphi$. □

Soundness and Completeness

The mutual relationship between model-theoretic and proof-theoretic consequence relations is an important topic.

Definition 1.1.18. Let L be a set, $\vDash$ be a semantic consequence relation on L and $\vdash$ be a (finitary) syntactic consequence relation on L. We say that:

- the proof-system used to define $\vdash$ is *weakly sound* for the semantics used to define $\vDash$ when, for every $\varphi \in \mathrm{L}$, if $\vdash \varphi$ then $\vDash \varphi$;

- the proof-system used to define $\vdash$ is *(strongly) sound* for the semantics used to define $\vDash$ when, for every $\Phi \subseteq \mathrm{L}$ and $\varphi \in \mathrm{L}$, if $\Phi \vdash \varphi$ then $\Phi \vDash \varphi$;

- the proof-system used to define $\vdash$ is *weakly complete* for the semantics used to define $\vDash$ when, for every $\varphi \in \mathrm{L}$, if $\vDash \varphi$ then $\vdash \varphi$;

- the proof-system used to define $\vdash$ is *(strongly) complete* for the semantics used to define $\vDash$ when, for every $\Phi \subseteq \mathrm{L}$ and $\varphi \in \mathrm{L}$, if $\Phi \vDash \varphi$ then $\Phi \vdash \varphi$.

When the semantics used to define $\vDash$ is clear from context, we simply say that the proof-system used to define $\vdash$ is weakly sound (resp. sound, weakly complete, complete).

1.1.3 Examples

We stressed that along this dissertation we explore several logics. In particular, the following sections are a survey of the logics that more directly influenced our work. For the sake of illustration, we now present classical propositional logic.

Example 1.1.19 (Classical propositional logic)**.** Let $\mathcal{P}$ be a set of *propositional symbols* and C be the set of classical connectives composed by the 1-ary connective $\neg$ and the 2-ary connectives $\wedge, \vee, \rightarrow$. According to definition 1.1.1, the language of classical propositional logic (CPL) is the set $\mathrm{L}_{\mathsf{CPL}}$ of *propositional formulas* defined inductively by

$$\mathrm{L}_{\mathsf{CPL}} ::= \mathcal{P} \mid \neg \mathrm{L}_{\mathsf{CPL}} \mid \mathrm{L}_{\mathsf{CPL}} \wedge \mathrm{L}_{\mathsf{CPL}} \mid \mathrm{L}_{\mathsf{CPL}} \vee \mathrm{L}_{\mathsf{CPL}} \mid \mathrm{L}_{\mathsf{CPL}} \rightarrow \mathrm{L}_{\mathsf{CPL}} \,.$$

The semantics for CPL is defined over a class $\{0,1\}^{\mathcal{P}}$ of all possible binary *valuations* over the propositional symbols $\mathcal{P}$. The value 0 should stand for falsity and 1 for truth. A valuation $v \in \{0,1\}^{\mathcal{P}}$ is extended to the set of propositional formulas, $\overline{v} : \mathrm{L}_{\mathsf{CPL}} \rightarrow \{0,1\}$, inductively as follows:

- $\overline{v}(p) = v(p)$;
- $\overline{v}(\neg\varphi) = 1 - \overline{v}(\varphi)$;
- $\overline{v}(\varphi_1 \rightarrow \varphi_2) = 1 + \overline{v}(\varphi_1) \cdot \overline{v}(\varphi_2) - \overline{v}(\varphi_1)$;
- $\overline{v}(\varphi_1 \wedge \varphi_2) = \overline{v}(\varphi_1) \cdot \overline{v}(\varphi_2)$;
- $\overline{v}(\varphi_1 \vee \varphi_2) = \overline{v}(\varphi_1) + \overline{v}(\varphi_2) - \overline{v}(\varphi_1) \cdot \overline{v}(\varphi_2)$,

for each $p \in \mathcal{P}$, $\varphi, \varphi_1, \varphi_2 \in \mathrm{L}_{\mathsf{CPL}}$. We abuse notation and denote $\overline{v}$ by v.

The satisfaction system that leads to the semantic consequence relation of CPL is the triple $\langle \mathrm{L}_{\mathsf{CPL}}, \{0,1\}^{\mathcal{P}}, \Vdash_{\mathsf{CPL}} \rangle$, where the satisfaction relation $\Vdash_{\mathsf{CPL}} \subseteq (\{0,1\}^{\mathcal{P}} \times \mathrm{L}_{\mathsf{CPL}})$ is defined as follows:

$$v \Vdash_{\mathsf{CPL}} \varphi \text{ iff } v(\varphi) = 1.$$

Taking into account the usual abbreviations for conjunction and disjunction ($\varphi_1 \wedge \varphi_2$ abbr. $\neg(\varphi_1 \rightarrow \neg\varphi_2)$ and $\varphi_1 \vee \varphi_2$ abbr. $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$), for the syntactic consequence relation we consider the deductive system $\mathcal{H}_{\mathsf{CPL}}$ consisting of axioms:

CPL1 $\varphi_1 \rightarrow (\varphi_2 \rightarrow \varphi_1)$,

CPL2 $(\varphi_1 \rightarrow (\varphi_2 \rightarrow \varphi_3)) \rightarrow ((\varphi_1 \rightarrow \varphi_2) \rightarrow (\varphi_1 \rightarrow \varphi_3))$,

CPL3 $(\neg\varphi_1 \rightarrow \neg\varphi_2) \rightarrow (\varphi_2 \rightarrow \varphi_1)$;

and additionally the inference rules of *modus ponens*:

MP $\frac{\varphi_1 \quad \varphi_1 \rightarrow \varphi_2}{\varphi_2}$,

for every $\varphi_1, \varphi_2, \varphi_3 \in \mathrm{L}_{\mathsf{CPL}}$.

The syntactic consequence relation of CPL is, then, the deducibility relation $\vdash_{\mathcal{H}_{\mathsf{CPL}}}$ obtained from the deductive system $\mathcal{H}_{\mathsf{CPL}}$, as stated in definition 1.1.16.

Soundness and completeness results for $\mathcal{H}_{\mathsf{CPL}}$ are well known (see for instance [81]), so both the semantic and the syntactic formulations of consequence relations for the language $\mathrm{L}_{\mathsf{CPL}}$ coincide: $\Vdash_{\mathsf{CPL}} = \vdash_{\mathcal{H}_{\mathsf{CPL}}}$. Having said this, CPL is the logic $\langle \mathrm{L}_{\mathsf{CPL}}, \vdash_{\mathcal{H}_{\mathsf{CPL}}} \rangle$. △

1.2 Probabilistic Logic

Probabilistic logic aims at dealing with probabilistic reasoning and with the underlying uncertainty in formal deductive proofs. Throughout the years, much effort was spent by a number of mathematicians and logicians (as J. Bernoulli, J. H. Lambert, A. De Morgan, G. Boole, C. S. Peirce, D. Scott, P. Krauss, T. Hailperin and N.J. Nilsson) on trying to formalize the reasoning about probabilities. In 1990, Fagin, Halpern and Megiddo presented a logic to reason explicitly about probabilities [58]. Their extensive work encompasses the measurable and the non-measurable cases. In this Section, we survey the measurable case.

Reasoning about probabilities is one of the most fundamental issues of our work, so we pay special attention to this topic. There are a number of techniques that are common to the reasoning underlying probabilities, so we profit to establish the general framework and notation in order for the remaining text to be consistent and reader-friendly.

1.2.1 Syntax and Semantics

The probabilistic logic introduced by Fagin, Halpern and Megiddo [58] relies on fixing a set $\mathcal{P}$ of propositional symbols and considering the set of propositional formulas $\mathrm{L}_{\mathsf{CPL}}$. The probabilistic propositional formulas are defined through a *weight term* language WTerm consisting of linear probabilistic terms with rational coefficients defined by the following grammar:

$$\mathsf{WTerm} ::= \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L}_{\mathsf{CPL}}) + \cdots + \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L}_{\mathsf{CPL}}) \enspace ,$$

which is used to define the set WAt of *weighted atoms* as follows:

$$\mathsf{WAt} ::= \mathsf{WTerm} \geq \mathbb{Q} \enspace .$$

The language of the logic consists of the following set $\mathsf{Prob}_{\mathsf{CPL}}$ of *probabilistic propositional formulas*:

$$\mathsf{Prob}_{\mathsf{CPL}} ::= \mathsf{WAt} \mid \neg \mathsf{Prob}_{\mathsf{CPL}} \mid \mathsf{Prob}_{\mathsf{CPL}} \wedge \mathsf{Prob}_{\mathsf{CPL}} \enspace .$$

Notice that in [58] the weighted terms involve integer coefficients, however this formulation is equivalent as we can always clear denominators. The usual abbreviations are accepted: $-q{\cdot}w$ abbr. $(-q)\cdot w$, $w_1 \geq w_2 + q$ abbr. $w_1 - w_2 \geq q$, $w < q$ abbr. $\neg(w \geq q)$, $w \leq q$ abbr. $-w \geq -q$, $w > q$ abbr. $-w < -q$, $w = q$ abbr. $w \leq q \wedge w \geq q$, $q_1 \leq w \leq q_2$ abbr. $w \geq q_1 \wedge w \leq q_2$, where $q, q_1, q_2 \in \mathbb{Q}, w, w_1, w_2 \in \mathsf{WTerm}$. The *propositional true* $\top$ is defined as an abbreviation for $p \vee \neg p$ for some fixed propositional symbol $p \in \mathcal{P}$ and the *propositional false* $\bot$ abbreviates $\neg\top$.

To interpret probabilistic propositional formulas, we consider the set of all valuations over $\mathcal{P}$ and endow it with a probability distribution. Let Π denote the set of all probability distributions over valuations in $\{0,1\}^{\mathcal{P}}$.

Definition 1.2.1. The *satisfaction relation* $\Vdash_{\mathsf{PrCPL}} \subseteq (\Pi \times \mathsf{Prob}_{\mathsf{CPL}})$ is defined inductively as follows:

- $\pi \Vdash_{\mathsf{PrCPL}} q_1 \cdot \Pr(\varphi_1) + \cdots + q_\ell \cdot \Pr(\varphi_\ell) \geq q$ iff $\sum\limits_{i=1}^{\ell} \left(q_i \left(\sum\limits_{v \in \{0,1\}^{\mathcal{P}}} (v(\varphi_i) \cdot \pi(v)) \right) \right) \geq q$;
- $\pi \Vdash_{\mathsf{PrCPL}} \neg f$ iff $\pi \nVdash_{\mathsf{PrCPL}} f$;
- $\pi \Vdash_{\mathsf{PrCPL}} f \wedge g$ iff $\pi \Vdash_{\mathsf{PrCPL}} f$ and $\pi \Vdash_{\mathsf{PrCPL}} g$,

for each probability distribution $\pi \in \Pi$, $\ell \geq 1$, $\varphi_1, \ldots, \varphi_\ell \in \mathrm{L}_{\mathsf{CPL}}$, $q_1, \ldots, q_\ell, q \in \mathbb{Q}$, $f, g \in \mathsf{Prob}_{\mathsf{CPL}}$. A probability distribution $\pi \in \Pi$ is said to satisfy $f \in \mathsf{Prob}_{\mathsf{CPL}}$ if $\pi \Vdash_{\mathsf{PrCPL}} f$.

1.2.2 Deductive System

The syntactic consequence relation is the consequence relation $\vdash_{\mathcal{H}_{\mathsf{PrCPL}}}$ obtained from the deductive system $\mathcal{H}_{\mathsf{PrCPL}}$ consisting of axioms:

C1$_{\mathsf{PrCPL}}$ $f_1 \to (f_2 \to f_1)$,
C2$_{\mathsf{PrCPL}}$ $(f_1 \to (f_2 \to f_3)) \to ((f_1 \to f_2) \to (f_1 \to f_3))$,
C3$_{\mathsf{PrCPL}}$ $(\neg f_1 \to \neg f_2) \to (f_2 \to f_1)$,

P1$_{\mathsf{PrCPL}}$ $\mathsf{Pr}(\varphi) \geq 0$,
P2$_{\mathsf{PrCPL}}$ $\mathsf{Pr}(\varphi_1 \wedge \varphi_2) + \mathsf{Pr}(\varphi_1 \wedge \neg\varphi_2) - \mathsf{Pr}(\varphi_1) = 0$,
P3$_{\mathsf{PrCPL}}$ $\mathsf{Pr}(\varphi_1) = \mathsf{Pr}(\varphi_2)$ if $\vdash_{\mathsf{CPL}} (\varphi_1 \leftrightarrow \varphi_2)$,
P4$_{\mathsf{PrCPL}}$ $\mathsf{Pr}(\top) = 1$,

I1$_{\mathsf{PrCPL}}$ $w \geq q \vee w \leq q$,
I2$_{\mathsf{PrCPL}}$ $w \geq q_1 \to w > q_2$, if $q_1 > q_2$,
I3$_{\mathsf{PrCPL}}$ $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \leftrightarrow q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) + 0 \cdot \mathsf{Pr}(\varphi_{\ell+1}) \geq q$,

I4$_{\mathsf{PrCPL}}$ $((q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q_0) \wedge (q'_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q'_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q'_0)) \to$
$\to ((q_1 + q'_1) \cdot \mathsf{Pr}(\varphi_1) + \cdots + (q_\ell + q'_\ell) \cdot \mathsf{Pr}(\varphi_\ell) \geq q_0 + q'_0)$,

I5$_{\mathsf{PrCPL}}$ $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \to (q' \cdot q_1) \cdot \mathsf{Pr}(\varphi_1) + \cdots + (q' \cdot q_\ell) \cdot \mathsf{Pr}(\varphi_\ell) \geq (q' \cdot q)$,

I6$_{\mathsf{PrCPL}}$ $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \leftrightarrow q_{i_1} \cdot \mathsf{Pr}(\varphi_{i_1}) + \cdots + q_{i_\ell} \cdot \mathsf{Pr}(\varphi_{i_\ell}) \geq q$;

and additionally the inference rules of *modus ponens*:

MP$_{\mathsf{PrCPL}}$ $\frac{f_1 \quad f_1 \to f_2}{f_2}$,

for every $f_1, f_2, f_3 \in \mathsf{Prob}_{\mathsf{CPL}}$, integer ℓ with $\ell \geq 0$, $\varphi, \varphi_1, \varphi_2, \ldots, \varphi_\ell, \varphi_{\ell+1} \in \mathrm{L}_{\mathsf{CPL}}$, $q, q', q_0, q_1, ..., q_\ell$, $q'_0, q'_1, \ldots, q'_\ell \in \mathbb{Q}$, $q' > 0$ and for every permutation $(i_1 \cdots i_\ell)$ of $(1 \cdots \ell)$.

The deductive system $\mathcal{H}_{\mathsf{PrCPL}}$ reflects the three main components of this probabilistic logic: it deals with classical reasoning, reasoning about probabilities, and reasoning about linear inequalities. We emphasize that in [58] it was shown that axioms I1$_{\mathsf{PrCPL}}$-I6$_{\mathsf{PrCPL}}$, C1$_{\mathsf{PrCPL}}$-C3$_{\mathsf{PrCPL}}$, and inference rule MP$_{\mathsf{PrCPL}}$ constitute a sound and complete axiomatization for reasoning about inequalities.

1.2.3 Soundness and Completeness

Now, we aim to recall the main techniques presented in [58] to prove that the deductive system $\mathcal{H}_{\mathsf{PrCPL}}$ is weakly complete. These will also constitute one of our main tools for dealing with probabilities afterwards.

Theorem 1.2.2 ([58]). *$\mathcal{H}_{\mathsf{PrCLP}}$ is sound and weakly complete.*

Sketch of the proof: The proof of soundness is straightforward, so we proceed by summarizing the main steps in the proof of completeness. The details may be found in [58].

Recall that we want to show that: if $\vDash_{\mathcal{H}_{\mathsf{PrCPL}}} f$ then $\vdash_{\mathcal{H}_{\mathsf{PrCPL}}} f$. The proof of completeness follows by contraposition. Hence, assume that $\nvdash_{\mathcal{H}_{\mathsf{PrCPL}}} f$ and try to find a model for $\neg f$. We use the classical feature of the logic to reduce $\neg f$ to an equivalent formula in *disjunctive normal form*, i.e., into a formula of the form $f_1 \vee \ldots \vee f_r$, where each f_i is a probabilistic propositional formula that is a conjunction of one or more weighted atoms or their negations. Since $\neg f$ is consistent, let $g^i_1 \wedge \ldots \wedge g^i_r \wedge \neg g^i_{r+1} \wedge \ldots \wedge \neg g^i_{r+s}$ represent a consistent disjunct f_i of $f_1 \vee \ldots \vee f_r$, where $g^i_1, \ldots, g^i_r, g^i_{r+1}, \ldots, g^i_{r+s} \in \mathsf{WAt}$, whose existence is guaranteed by the Lindenbaum's Lemma (Lemma 1.1.8). Then, let $P_0 = \{p_1, \ldots, p_n\}$ be the set of all propositional symbols occurring in f_i and consider the set $\Theta = \left\{ \bigwedge_{p \in Q} p \wedge \bigwedge_{q \in P_0 \setminus Q} \neg q \mid Q \subseteq P_0 \right\}$. Fix an enumeration of the 2^n elements of Θ. Once proved that $\mathsf{Pr}(\varphi) = \sum_{\theta \in \Theta \text{ st } \theta \to \varphi} \mathsf{Pr}(\theta)$, one can instantiate each weighted term $\mathsf{Pr}(\varphi)$ occurring in f_i with the weighted term $\sum_{\theta \in \Theta \text{ st } \theta \to \varphi} \mathsf{Pr}(\theta)$ and, clearing denominators, f_i turns out to be equivalent to the probabilistic propositional formula that results from the conjunction of the following formulas:

$$\begin{cases} a_{1,1} \cdot \mathsf{Pr}(\theta_1) + \cdots + a_{1,2^n} \cdot \mathsf{Pr}(\theta_{2^n}) \geq a_1 \\ \qquad\qquad \vdots \\ a_{r,1} \cdot \mathsf{Pr}(\theta_1) + \cdots + a_{r,2^n} \cdot \mathsf{Pr}(\theta_{2^n}) \geq a_r \\ a'_{1,1} \cdot \mathsf{Pr}(\theta_1) + \cdots + a'_{1,2^n} \cdot \mathsf{Pr}(\theta_{2^n}) < a'_1 \\ \qquad\qquad \vdots \\ a'_{s,1} \cdot \mathsf{Pr}(\theta_1) + \cdots + a'_{s,2^n} \cdot \mathsf{Pr}(\theta_{2^n}) < a'_s \\ \mathsf{Pr}(\theta_1) + \cdots + \mathsf{Pr}(\theta_{2^n}) = 1 \\ \mathsf{Pr}(\theta) \geq 0 & \text{for all } \theta \in \Theta \end{cases} \tag{1.1}$$

for some integers $a_i, a'_j, a_{i,k}, a'_{j,k}$ with $i \in \{1, \ldots, r\}, j \in \{1, \ldots, s\}, k \in \{1, \ldots, 2^n\}$. But since the probabilities can be assigned independently to each element of Θ, we can use the result of soundness and completeness for the axioms of inequality (for details, see Section 4 of [58]) and conclude that (1.1) is unsatisfiable if and only if it is inconsistent. But the inconsistency of (1.1) would lead to the inconsistency of f, so we conclude that (1.1) is satisfiable, hence $\neg f$ is also satisfiable. □

Let us denote by PrCPL the probabilistic logic $\langle \mathsf{Prob}_{\mathsf{CPL}}, \vdash_{\mathcal{H}_{\mathsf{PrCPL}}} \rangle$.

1.2.4 Satisfiability and Complexity

The reasoning carried out in the proof of completeness let Fagin et al. take one step further to prove a *small-model theorem*, which states that a satisfiable probabilistic propositional formula f is satisfiable in a small model, meaning that f is certainly satisfiable in a model whose probability distribution assigns at most $|f|$ non-zero values of probabilities to valuations. This result follows as an immediate consequence of the famous result from linear programming presented in [35], which we state now.

Lemma 1.2.3 ([35, 58]). *If a system of ℓ linear inequalities with integer coefficients has a non-negative solution, then it has a non-negative solution with at most ℓ positive entries.*

The small-model theorem is naturally followed by a complexity result for the satisfiability problem. In general, the satisfiability problems consist in deciding the existence of a model for a given formula. For the probabilistic logic that we present in this section, the satisfiability problem consists in deciding the existence of a probability distribution satisfying a probabilistic propositional formula. Once proved the small-model theorem, the complexity result for the satisfiability problem of this logic emerges (details can be found in [58]).

Theorem 1.2.4 ([58]). *The satisfiability problem for the probabilistic logic* PrCPL *is* NP*-complete.*

1.3 Equational Logic

Equality is probably one of the most familiar symbols in mathematics. This concept has a long history that goes from the ancient association to the result of a calculation to the major role that it plays in computation. The need for a rigorous treatment of equality rapidly emerged and motivated the formalization of equational reasoning.

The first step towards equational logic is conferred to the Peano arithmetic that axiomatize equality on natural numbers [95]. It was followed by several results from Russel, Tarski and Church in type theory and great developments in universal algebra and equational logic due to Birkhoff, Grätzer, Cohn and Tarski [24, 38, 67, 105].

1.3.1 Terms, Equations, and Algebras

In general, terms are constructed from generators and function symbols. To clarify the context in which we choose the function symbols, let us defined what is intended to be a *signature.*

Definition 1.3.1. A *signature* is a $\mathbb{N}$-indexed family of countable sets F_n of *function symbols* of *arity* n, $\mathrm{F} = \{\mathrm{F}_n\}_{n\in\mathbb{N}}$. The elements of F_0 are also called *constant symbols.*

Definition 1.3.2. Let F be a signature and G be a set of *generators.* The set $T_{\mathrm{F}}(G)$ of *terms over* G is defined inductively by:

- $G \subseteq T_{\mathrm{F}}(G)$;
- if $t_1, \ldots, t_n \in T_{\mathrm{F}}(G)$ and $f \in \mathrm{F}_n$, then $f(t_1 \ldots, t_n) \in T_{\mathrm{F}}(G)$.

Under some generality, we defined terms over a set of generators because later we will be interested in distinguishing *algebraic terms* defined over a set of variables, from *nominal terms* defined over a set of names. Throughout the text we drop the subscript F when it is clear from the context.

Definition 1.3.3. Let F be a signature and G be a set of generators. The set $\mathsf{gen}(t)$ of *generators* occurring in a term $t \in T(G)$ is defined inductively by:

- $\mathsf{gen}(g) = \{g\}$, for each $g \in G$;
- $\mathsf{gen}(f(t_1 \ldots, t_n)) = \bigcup_{i=1}^{n} \mathsf{gen}(t_i)$, for each $n \in \mathbb{N}$, $f \in \mathrm{F}_n$ and $t_1, \ldots, t_n \in T(G)$.

Notice that throughout the text, the set of generators occurring in a term will be renamed depending on the set of generators considered.

Definition 1.3.4. Let F be a signature and G be a set of generators. The set $\mathsf{subterms}(t)$ of *subterms* of a term $t \in T(G)$ is defined inductively by:

- $\mathsf{subterms}(g) = \{g\}$, for each $g \in G$;
- $\mathsf{subterms}(f(t_1 \ldots, t_n)) = \{f(t_1 \ldots, t_n)\} \cup \bigcup_{i=1}^{n} \mathsf{subterms}(t_i)$, for each $n \in \mathbb{N}$, $f \in \mathrm{F}_n$ and $t_1, \ldots, t_n \in T(G)$.

We extend the notion of subterm to sets of terms. Given a set of terms $T_0 \subseteq T(G)$, we define the set of subterms occurring in T_0 as $\mathsf{subterms}(T_0) = \bigcup_{t \in T_0} \mathsf{subterms}(t)$.

Definition 1.3.5. Let F be a signature and G be a set of generators. The *length of a term* $t \in T(G)$ is denoted by $|t|$ and defined inductively by:

- $|g| = 1$, for each $g \in G$;
- $|f(t_1 \ldots, t_n)| = 1 + |t_1| + \ldots + |t_n|$, for each $n \in \mathbb{N}$, $f \in \mathrm{F}_n$ and $t_1, \ldots, t_n \in T(G)$.

The following proposition is an immediate corollary of the previous definitions.

Proposition 1.3.6. *Let* F *be a signature and G be a set of generators. The number of subterms of a term $t \in T(G)$ is at most $|t|$.*

Definition 1.3.7. Let F be a signature and G_1, G_2 be any sets. A *substitution* from G_1 to $T_{\mathrm{F}}(G_2)$ is a function $\sigma : G_1 \to T_{\mathrm{F}}(G_2)$. The set of all substitutions from G_1 to $T_{\mathrm{F}}(G_2)$ is denoted by $T_{\mathrm{F}}(G_2)^{G_1}$. Any substitution can be extended to the set of terms over G_1, $\overline{\sigma} : T_{\mathrm{F}}(G_1) \to T_{\mathrm{F}}(G_2)$ as follows:

- $\overline{\sigma}(g) = \sigma(g)$ for each $g \in G_1$;
- $\overline{\sigma}(f(t_1, \ldots, t_n)) = f(\overline{\sigma}(t_1), \ldots, \overline{\sigma}(t_n))$, for every $n \in \mathbb{N}$, $f \in \mathrm{F}_n$ and $t_1, \ldots, t_n \in T_{\mathrm{F}}(G_1)$.

We abuse notation and use σ to denote $\overline{\sigma}$. A term $t \in T_{\mathrm{F}}(G_2)$ is called an *instance* of a term $s \in T_{\mathrm{F}}(G_1)$ if there exists a substitution $\sigma \in T_{\mathrm{F}}(G_2)^{G_1}$ such that $\sigma(s) = t$.

Now we proceed defining the atoms of interest for the equational logic.

Definition 1.3.8. Let F be a signature and G be a set. An *equation* is a pair $(t_1, t_2) \in T(G) \times T(G)$. We represent an equation as $t_1 \approx t_2$. The set of all equations over G is denoted by $\mathsf{Eq}_{\mathrm{F}}(G)$. We drop the subscript F when it is clear from context.

The set of subterms of an equation is defined as $\mathsf{subterms}(t_1 \approx t_2) = \{t_1, t_2\}$ and the set of generators occurring in $t_1 \approx t_2$ is the set $\mathsf{gen}(t_1 \approx t_2) = \mathsf{gen}(t_1) \cup \mathsf{gen}(t_2)$.

Definition 1.3.9. Let F be a signature. An F-*algebra* is a pair $\mathbb{A} = \langle A, (\cdot)^{\mathbb{A}} \rangle$ where:

- the *carrier* A is a non-empty set;
- the *interpretation* of function symbols $(\cdot)^{\mathbb{A}}$ is such that, for each $f \in \mathrm{F}_n$, $f^{\mathbb{A}} : A^n \to A$ is an operation on A.

Definition 1.3.10. Let F be a signature and G be a set of generators. A *term algebra* is the F-algebra $\mathbb{T}_{\mathrm{F}}(G)$ whose carrier is $T_{\mathrm{F}}(G)$ and the interpretation of terms is such that, for each $f \in \mathrm{F}_n$, $f^{\mathbb{T}_{\mathrm{F}}(G)}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$.

Definition 1.3.11. Let F be a signature and $\mathbb{A}$ be an F-algebra with carrier A. An equivalence relation $\equiv \subseteq (A \times A)$ is a *congruence relation* if and only if for every $n \in \mathbb{N}$ and $f \in \mathrm{F}_n$,

$$f^{\mathbb{A}}(a_1, \dots, a_n) \equiv f^{\mathbb{A}}(a_1', \dots, a_n') \text{ whenever } a_1 \equiv a_1' \dots, a_n \equiv a_n'.$$

The *quotient algebra* $\mathbb{A}_{/\equiv}$ has as carrier the set of equivalence classes $\{[a]_{\equiv} \mid a \in A\}$ and the interpretation of function symbols is such that for each $f \in \mathrm{F}_n$, $f^{\mathbb{A}_{/\equiv}}([a_1]_{\equiv}, \dots, [a_n]_{\equiv}) = [f^{\mathbb{A}}(a_1, \dots, a_n)]_{\equiv}$.

1.3.2 Syntax and Semantics

The equational logic [13,56,61,71] relies on fixing a signature F and a countable set of *variables* X. Since the generators for terms are simply variables, $\mathsf{vars}(t)$ stands for the set of variables occurring in a term $t \in T(X)$.

The language of the logic consists of the set $\mathsf{Eq}(X)$ of *equations* over X:

$$\mathsf{Eq}(X) ::= T(X) \approx T(X) \enspace .$$

The semantics for equational logic is defined over a class $\mathcal{A}$ of F-algebras. Each of them should be provided of a set of assignments of values to variables.

Definition 1.3.12. Let $\mathbb{A}$ be an F-algebra with carrier set A. An *assignment* is a function $\sigma : X \to A$. The set of all assignments is denoted by A^X. An assignment can be extended to the set of terms, $[\![\cdot]\!]_{\mathbb{A}}^{\sigma} : T(X) \to A$ as follows:

- $[\![x]\!]_{\mathbb{A}}^{\sigma} = \sigma(x)$ for each $x \in X$;
- $[\![f(t_1, \dots, t_n)]\!]_{\mathbb{A}}^{\sigma} = f^{\mathbb{A}}([\![t_1]\!]_{\mathbb{A}}^{\sigma}, \dots, [\![t_n]\!]_{\mathbb{A}}^{\sigma})$, for every $n \in \mathbb{N}$, $f \in \mathrm{F}_n$, $t_1, \dots, t_n \in T(X)$.

Definition 1.3.13. Let $\mathbb{A}$ be an F-algebra with carrier set A and $\sigma \in A^X$ an assignment. We say that an equation $t_1 \approx t_2$ is *satisfiable in* $\mathbb{A}$ *provided with* σ, and write $\mathbb{A}, \sigma \Vdash t_1 \approx t_2$, if $[\![t_1]\!]_{\mathbb{A}}^{\sigma} = [\![t_2]\!]_{\mathbb{A}}^{\sigma}$.

Definition 1.3.14. Let $\mathcal{A}$ be a class of F-algebras. The satisfaction relation $\Vvdash \subseteq (\mathcal{A} \times \mathsf{Eq}(X))$ is such that $\mathbb{A} \Vvdash t_1 \approx t_2$ if and only if $\mathbb{A}, \sigma \Vvdash t_1 \approx t_2$ for every assignment $\sigma \in A^X$. The satisfaction relation is extended to sets of equations as usual. The *semantic entailment* $\vDash^{(\Vvdash)}_{\mathcal{A}} \subseteq (\wp(\mathsf{Eq}(X)) \times \mathsf{Eq}(X))$ is defined as follows: given $\Gamma \subseteq \mathsf{Eq}(X)$ and $t_1 \approx t_2 \in \mathsf{Eq}(X)$, $\Gamma \vDash^{(\Vvdash)}_{\mathcal{A}} t_1 \approx t_2$ provided that every $\mathbb{A} \in \mathcal{A}$ satisfies $t_1 \approx t_2$ whenever it satisfies Γ.

We now define an important class of algebras that consists of *varieties* [13].

Definition 1.3.15. Let Γ be a set of equations. An F-algebra $\mathbb{A}$ is said to be a *model* of Γ if $\mathbb{A}$ satisfies Γ, $\mathbb{A} \Vvdash \Gamma$. The class of all models of Γ is called the Γ-*variety.*

The concept of an equational theory generated by a set of equations arises naturally [13,97].

Definition 1.3.16. The *equational theory* induced by a set of equations $\Gamma \subseteq \mathsf{Eq}(X)$ is the set $\mathsf{Th}(\Gamma)$ of equations satisfied by all the models in the Γ-variety $\mathcal{A}_\Gamma$,

$$\mathsf{Th}(\Gamma) = \{t_1 \approx t_2 \mid \Gamma \vDash^{(\Vvdash)}_{\mathcal{A}_\Gamma} t_1 \approx t_2\}.$$

1.3.3 Deductive System

The syntactic consequence relation on $\mathsf{Eq}(X)$ is the consequence relation $\vdash_{\mathcal{H}_{\mathsf{EqL}}}$ obtained from the deductive system $\mathcal{H}_{\mathsf{EqL}}$ consisting of the following inference rules:

Ref $$\frac{}{t \approx t}$$

Sym $$\frac{t_1 \approx t_2}{t_2 \approx t_1}$$

Trans $$\frac{t_1 \approx t_2 \qquad t_2 \approx t_3}{t_1 \approx t_3}$$

Cong $$\frac{t_1 \approx t_1' \quad \cdots \quad t_n \approx t_n'}{f(t_1, \ldots, t_n) \approx f(t_1', \ldots, t_n')}$$

Sub $$\frac{t_1 \approx t_2}{\sigma(t_1) \approx \sigma(t_2)}$$

for every $t_1, t_2, t_3, \ldots, t_n, t_1', \ldots, t_n' \in T(X)$ and $\sigma \in T(X)^X$.

The equational logic can be seen a simple fragment of first order logic with equality devoid of quantifiers and connectives [78, 80, 97].

1.3.4 Soundness and Completeness

Soundness and completeness results for $\mathcal{H}_{\mathsf{EqL}}$ are well known (see for instance [13, 24]).

Theorem 1.3.17 ([24])**.** *$\mathcal{H}_{\mathsf{EqL}}$ is sound and complete.*

1.3.5 Extensions of Equational Logic

The notions that we have been developing and the deductive system that we presented can be generalized to more expressive frameworks in the equational context.

Equational Horn Logic

Our concern is to specify some equational properties and, in this matter, it turns out to be essential to get more expressivity. Even though this idea is amply discussed in the literature (see [56, 80]), it is extensively analyzed in [89].

A *Horn clause* over a set of variables X is an expression $(t_1 \approx t'_1\,, \ldots\,, t_k \approx t'_k \Rightarrow t \approx t')$, with $k \geq 0$ and t, t', t_1, ... , t_k, t'_1, ... , $t'_k \in T(X)$. A Horn clause is simply an equation when $k = 0$. We omit the enclosing parentheses when no ambiguities arise.

Often, in the literature, a Horn clause is also called a *conditional equation* (see, for instance, [89]). Kowalski motivated the importance of Horn clauses in detriment of clauses in general [72] stating that: "most of the models for problem-solving which have been developed in artificial intelligence can be regarded as models for problems expressed by means of Horn clauses."([72], p.17).

The interpretation of a Horn clause in an F-algebra $\mathbb{A}$ with respect to $\sigma \in A^X$ is defined by: $\mathbb{A}, \sigma \Vdash (t_1 \approx t'_1, \ldots, t_k \approx t'_k \Rightarrow t \approx t')$ if whenever $[\![t_i]\!]^\sigma_{\mathbb{A}} = [\![t'_i]\!]^\sigma_{\mathbb{A}}$ for each $1 \leq i \leq k$ then $[\![t]\!]^\sigma_{\mathbb{A}} = [\![t']\!]^\sigma_{\mathbb{A}}$. An algebra $\mathbb{A}$ satisfies a Horn clause if it is satisfied by $\mathbb{A}$ along with each $\sigma \in A^X$. More generally, a Horn clause is satisfied in a class of algebras $\mathcal{A}$ if it is satisfied in every $\mathbb{A} \in \mathcal{A}$. The notion of satisfaction can be generalized to sets of Horn clauses and the semantic entailment is defined as usual.

Given a finite set of Horn clauses Γ, an F-algebra $\mathbb{A}$ is said to be a *model* of Γ if $\mathbb{A}$ satisfies Γ, $\mathbb{A} \Vdash \Gamma$. The class of all models of Γ is called a Γ-*quasivariety.*

Consider the deductive system $\mathcal{H}_{\mathsf{Horn}}$ composed by the inference rules of $\mathcal{H}_{\mathsf{EqL}}$ as axioms in the implication form:

Ref $t \approx t$

Sym $t_1 \approx t_2 \Rightarrow t_2 \approx t_1$

Trans $t_1 \approx t_2, t_2 \approx t_3 \Rightarrow t_1 \approx t_3$

Cong $t_1 \approx t'_1, \ldots, t_n \approx t'_n \Rightarrow f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$

and additionally by the inference rule Sub and also the Cut rule:

$$\textsf{\textbf{Sub}}\quad \frac{t_1 \approx t_2}{\sigma(t_1) \approx \sigma(t_2)}$$

$$\textsf{\textbf{Cut}}\quad \frac{\Gamma \cup \{t_1' \approx t_2'\} \Rightarrow t_1 \approx t_2 \quad \Delta \Rightarrow t_1' \approx t_2'}{\Gamma \cup \Delta \Rightarrow t_1 \approx t_2}$$

for every $t_1, t_2, t_3, \ldots, t_n, t_1', t_2' \ldots, t_n' \in T(X)$, $\Gamma, \Delta \subseteq T(X)$ and $\sigma \in T(X)^X$.

Soundness and completeness of this deductive system is a well-known result (see, for instance, [89])

Theorem 1.3.18 ([89]). *$\mathcal{H}_{\textsf{Horn}}$ is sound and complete.*

Equational Logic and Rewriting

The marriage between logic and computer science has revealed the need to give a computational dimension to the algebraic capacity of rewriting expressions. Namely, the intuition underlying the simplification of an expression, which cannot be formalized using the standard notion of equality, naturally emerges using the notion of *rewriting* as an oriented equality.

Since our motivation relies, precisely, on equational theories arising from the algebraic requirements underlying security protocols, we are particularly interested in equational theories generated by convergent rewriting systems. A lot of research has been done in this topic [56, 71, 89]. We mainly base this survey in rewriting systems on the extensive analysis of Baader and Nipkow in [13].

Given a set of variables X, a *rewriting system* R is a finite set of *rewrite rules* $\ell \to r$, where $\ell, r \in T(X)$ and it is often required that $\mathsf{vars}(r) \subseteq \mathsf{vars}(\ell)$. A rewrite rule is applied by replacing an instance of the left-hand side by the same instance of its right-hand side.

Given a rewriting system R and a set of generators G, the *rewriting relation* $\to_R \subseteq T(G) \times T(G)$ on $T(G)$ is the smallest relation such that:

- if $(\ell \to r) \in R$ and $\sigma : X \to T(G)$ is a substitution, then $\sigma(\ell) \to_R \sigma(r)$
- if $f \in \mathrm{F}_n$, $t_1, \ldots, t_n, t_i' \in T(G)$ and there exists $i \in \{1, \ldots, n\}$ such that $t_i \to_R t_i'$, then $f(t_1, \ldots, t_i, \ldots, t_n) \to_R f(t_1, \ldots, t_i', \ldots, t_n)$.

We denote by $\to_R^*$ the reflexive and transitive closure of $\to_R$.

A term $t \in T(G)$ is said to be *reducible* if there exists a term $t' \in T(G)$ such that $t \to_R t'$. A term $t \in T(G)$ is said to be in *normal form* if it is not reducible. A term t' is *a normal form of* t if t' is in normal form and $t \to_R^* t'$. We denote by $t{\downarrow}$ the normal form of $t \in T(G)$, when it exists.

A rewriting system R is *confluent* if, given $t \in T(G)$, $t \to_R^* t'$ and $t \to_R^* t''$ implies that there exists $t^* \in T(G)$ such that $t' \to_R^* t^*$ and $t'' \to_R^* t^*$. R is *terminating* if there exists no infinite rewriting sequence. R is *convergent* if it is confluent and terminating. If a rewriting system is convergent then any $t \in T(G)$ has a unique normal form (see [13]), i.e., there exists a term $t{\downarrow} \in T(G)$ such that $t \to_R^* t{\downarrow}$ and $t{\downarrow}$ is irreducible.

The *equational theory generated by a convergent rewriting system R* is the set of equations such that $t_1 \approx_R t_2$ if and only if $t_1{\downarrow} = t_2{\downarrow}$, also said to be a *convergent equational theory.* A convergent equational theory is known to always be decidable (see [13]).

A rewriting system R is said to be *subterm convergent* [3, 39] if $r \in \mathsf{subterms}(\ell) \cup \mathrm{F}_0$ and $\ell \notin \mathrm{F}_0$ for each rewrite rule $(\ell \to r) \in R$. An equational theory generated by a subterm convergent rewriting systems is called *subterm theory.*

Example 1.3.19. The sum (xor) of single bits can be characterized considering a signature $\mathrm{F}^{\mathrm{xor}}$ with three function symbols: $\mathsf{zero} \in \mathrm{F}_0^{\mathrm{xor}}$, $\mathsf{suc} \in \mathrm{F}_1^{\mathrm{xor}}$, $\oplus \in \mathrm{F}_2^{\mathrm{xor}}$, and the equational theory $\mathsf{Th}(\Gamma^{\mathrm{xor}})$ where

$$\Gamma^{\mathrm{xor}} = \{\mathsf{zero} \oplus x \approx x, \mathsf{suc}(x) \oplus y \approx x \oplus \mathsf{suc}(y), \mathsf{suc}(\mathsf{suc}(x)) \approx x\}.$$

Obviously, $\mathbb{Z}_2$ with the usual interpretations for zero, successor and sum modulo 2 satisfies Γ^{xor}. Furthermore, it must be clear that the rewriting system obtained by giving to each of the equations a left-to-right orientation is convergent. However, it is not subterm convergent due to the second equation. △

Equations and Domain Restrictions

In this subsection we are targeting our analysis to a wider algebraic reasoning. This is a subsection purely based on our information security motivation for this work and consists in extending the algebraic scope to the analysis of domain restrictions. Only in this way we will be able to model the reasoning of an attacker with cryptanalytic capabilities, as inspired by [83].

We motivate our domain restriction analysis on the previous Horn clause analysis. For this purpose, let us consider a signature F, a set of generators G and a finite set of *domain names* $\mathcal{D}$. We use $t \in \mathrm{D}$ (resp., $t \notin \mathrm{D}$) to represent the fact that a term $t \in T(G)$ belongs (resp., does not belong) to a domain $\mathrm{D} \in \mathcal{D}$. We dub the expression $t \in \mathrm{D}$ (resp., $t \notin \mathrm{D}$) a *positive* (resp., *negative*) *domain restriction*, and define the set of subterms of a (positive or negative) domain restriction $t \circledin \mathrm{D}$, with $\circledin \in \{\in, \notin\}$, as $\mathsf{subterms}(t \circledin \mathrm{D}) = \{t\}$. The set of generators of a domain restriction is the set $\mathsf{gen}(t \circledin \mathrm{D}) = \mathsf{gen}(t)$. Further, we use $\mathsf{DRes}(G)$ to denote the set of all positive domain restrictions over G.

A *domain clause* is an expression of the form $(t_1 \in \mathrm{D}_1, ..., t_{k_1} \in \mathrm{D}_{k_1} \Rightarrow t'_1 \textcircled{\in} \mathrm{D}'_1, ..., t'_{k_2} \textcircled{\in} \mathrm{D}'_{k_2})$, where the right-hand side is a non-empty sequence of (positive or negative) domain restrictions, i.e., $k_2 > 0$ and $\textcircled{\in} \in \{\in, \notin\}$. When $t'_1 = \cdots = t'_{k_2} = t$ and $t_1, \ldots, t_{k_1} \in \mathsf{subterms}(t)$, we say that the domain clause satisfies the *subterm property.* Again, we omit the enclosing parentheses when no ambiguities arise.

Note that the subterm property is satisfied by an interesting range of examples, as it basically means that a domain restriction on a term is only conditioned by domain restrictions of its subterms.

We define an algebraic domain interpretation as a pair $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$, where $\mathbb{A}$ is an F-algebra with carrier set A and $\mathrm{I}^{\mathbb{A}} : \mathcal{D} \to \wp(A)$ fixes an interpretation of domain names as subsets of A. Given an assignment $\sigma \in A^X$, the interpretation of domain clauses is defined, as expected, by: $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \sigma \Vvdash (t_1 \in \mathrm{D}_1, \ldots, t_{k_1} \in \mathrm{D}_{k_1} \Rightarrow t'_1 \textcircled{\in} \mathrm{D}'_1, \ldots, t'_{k_2} \textcircled{\in} \mathrm{D}'_{k_2})$ if whenever $[\![t_i]\!]^{\sigma}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}_i)$ for each $1 \leq i \leq k_1$ then $[\![t'_j]\!]^{\sigma}_{\mathbb{A}} \textcircled{\in} \mathrm{I}^{\mathbb{A}}(\mathrm{D}'_j)$ for some $1 \leq j \leq k_2$. An algebraic domain interpretation $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ satisfies a domain clause if it is satisfied by $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ along with each $\sigma \in A^X$. Moreover, a domain clause is satisfied in a class of algebraic domain interpretations $\mathcal{I}$ if it is satisfied by each $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}$.

Example 1.3.20. Let us extend Example 1.3.19 by introducing a couple of domain names, $\mathcal{D}^{\text{xor}} = \{\mathsf{even}, \mathsf{odd}\}$, which are intended to obey some domain clauses:

$$\Lambda^{\text{xor}} = \{\mathsf{zero} \in \mathsf{even}, (x \in \mathsf{even} \Rightarrow \mathsf{suc}(x) \in \mathsf{odd}), x \in \mathsf{odd} \Rightarrow \mathsf{suc}(x) \in \mathsf{even}, x \in \mathsf{odd} \Rightarrow x \notin \mathsf{even}\}.$$

Note that each domain clause in Λ_2 satisfies the subterm property, as the behaviour of terms is conditioned by restrictions on their subterms.

It is clear that the algebraic domain interpretation $(\mathbb{Z}_2, \mathrm{I}^{\mathbb{Z}_2})$, with $\mathrm{I}^{\mathbb{Z}_2}$ defined as $\mathrm{I}^{\mathbb{Z}_2}(\mathsf{even}) = \{0\}$ and $\mathrm{I}^{\mathbb{Z}_2}(\mathsf{odd}) = \{1\}$, satisfies Λ_2. △

Chapter 2

Equation-Based Classical Logic

In this chapter we propose and study an equation-based classical logic (EQCL) able to state and reason about equational constraints, by combining aspects of classical propositional logic, equational logic, and quantifiers. As we previously referred, this logic is mainly motivated by the reasoning required to perform static analysis of offline guessing attacks [18] and aims to capture the algebraic relationships between the messages exchanged within the context of a cryptographic protocol.

The logic is designed as a simple *global* classical logic built on top of a *local* equational basis. These two layers are permeated by a second-order-like quantification mechanism over *outcomes*. Intuitively, the attacker refers to messages using *names* whose concrete values are not important, but are gathered in a set of possible *outcomes*. The local layer allows us to reason about and define equational constraints on individual outcomes. At the global layer, we can state and reason about properties of the set of all possible outcomes. Interestingly, the quantification we use can be understood as an $S5$-like modality, which also explains why we will not need to consider nested quantifiers. EQCL bears important similarities with exogenous logics in the sense of [77], and with probabilistic logics as developed, for instance, in [58]. We provide a sound and complete deductive system for the logic, given a Horn-clause equational specification of the algebraic basis. We also show that the logic is decidable when the base equational theory can be given by means of a convergent rewriting system. Our decidability proof is actually more informative, as we develop a satisfiability procedure for our logic by means of a polynomial reduction to satisfiability for propositional classical logic. This strategy is useful as it enables building prototype tools for the logic using available SAT-solvers, and uses techniques that are similar to those used in the SMT literature [87]. As an application, we analyze offline guessing attacks to security protocols, where the equational basis specifies the algebraic properties of the cryptographic primitives.

The chapter is outlined as follows: in Section 2.1 we define the syntax and semantics of EqCL; in Section 2.2 we define a deductive system, whose soundness and completeness we prove in Section 2.3, assuming that we are given an equational specification of the algebraic basis; Section 2.4 is dedicated to showing, via a polynomial reduction to classical SAT, that EqCL is decidable whenever the equational base is given by means of a convergent rewriting system and, interestingly, it turns out that under such (not quite that restraining) restrictions, the complexity is not worse than for classical propositional logic; finally, in Section 2.5 we illustrate the usefulness of this logic with meaningful examples, namely related to the analysis of offline guessing attacks to security protocols. This work is published in [84] and is now presented with a more detailed analysis of the satisfiability and complexity results.

2.1 Syntax and Semantics

The logic EqCL relies on fixing a signature F, a set of variables X and class $\mathcal{A}$ of F-algebras. We also introduce a countable set of *names* N, distinct from variables.

Elements of $T(X)$ will be referred to as *algebraic terms* and $\mathsf{vars}(t)$ stands for the set of variables occurring in $t \in T(X)$. In the other hand, we dub elements of $T(N)$ as *nominal terms* and $\mathsf{names}(t)$ stands for the set of names that occur in $t \in T(N)$. Names can be thought of as being associated to values that are not made explicit. We call *outcome* to each possible concrete assignment of values to names.

The language of EqCL, designed in order to express equational constraints locally on each outcome, but also global properties of the set of all intended outcomes, is the set Glob defined by the following grammar:

$$\begin{array}{lcl} \mathsf{Glob} & ::= & \forall\mathsf{Loc} \mid \neg\mathsf{Glob} \mid \mathsf{Glob} \wedge \mathsf{Glob} \\ \mathsf{Loc} & ::= & \mathsf{Eq}(N) \mid \neg\mathsf{Loc} \mid \mathsf{Loc} \wedge \mathsf{Loc}\,. \end{array}$$

We abbreviate $\neg(t_1 \approx t_2)$ by $t_1 \not\approx t_2$ for any $t_1, t_2 \in T(N)$, and also use the usual abbreviations: $\psi_1 \vee \psi_2$ abbr. $\neg(\neg\psi_1 \wedge \neg\psi_2)$, $\psi_1 \to \psi_2$ abbr. $\neg\psi_1 \vee \psi_2$, $\psi_1 \leftrightarrow \psi_2$ abbr. $(\psi_1 \to \psi_2) \wedge (\psi_2 \to \psi_1)$, where either $\psi_1, \psi_2 \in \mathsf{Loc}$ or $\psi_1, \psi_2 \in \mathsf{Glob}$. Note that both the local and global languages are classical: the former with an equational basis and the later over local formulas instead of propositional symbols.

A *literal* is a global formula in $\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}$. We say that a global formula is in *disjunctive normal form* (DNF) if it is a disjunction of one or more conjunctions of literals. And it is in *conjunctive normal form* (CNF) if it is a conjunction of one or more disjunctions of literals. Elements of $\forall\mathsf{Loc}$ are referred to as (global) *atoms*.

We extend the notion of subterm to global formulas in a standard way. Similarly, we generalize the notion of names occurring in a nominal term to local and global formulas. We define the set of subformulas of either a local or a global formula ψ in the usual way and denote it by $\mathsf{subform}(\psi)$.

Given a nominal term $t_0 \in T(N)$, a set of names $\tilde{n} = \{n_1, \dots, n_k\} \subseteq N$ such that $\mathsf{names}(t_0) \subseteq \tilde{n}$ and $\tilde{t} = \{t_1, \dots, t_k\} \subseteq T(N)$ we denote by $[t_0]^{\tilde{n}}_{\tilde{t}}$ the nominal term obtained by replacing each occurrence of n_i by t_i, $i \in \{1, \dots, k\}$, i.e., $[t_0]^{\tilde{n}}_{\tilde{t}} = \sigma(t_0)$ where σ is a substitution such that $\sigma(n_i) = t_i$ for each i. This notion is easily extended to local formulas.

As explained above, names carry a form of undeterminedness, i.e., their values are fixed but we have no explicit knowledge about them. We will dub the possible concretizations of names by *outcomes*[1].

Definition 2.1.1. Given an F-algebra $\mathbb{A}$ with carrier set A, we define an *outcome* as a function $\rho : N \to A$. The set of all outcomes is denoted by A^N. The interpretation of terms $[\![\cdot]\!]^{\rho}_{\mathbb{A}} : T_{\mathrm{F}}(N) \to A$ is defined as usual.

Definition 2.1.2. Given an F-algebra $\mathbb{A}$ with carrier set A and an outcome $\rho \in A^N$, the *satisfiability of local formulas* is defined inductively by:

- $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} t_1 \approx t_2$ iff $[\![t_1]\!]^{\rho}_{\mathbb{A}} = [\![t_2]\!]^{\rho}_{\mathbb{A}}$,
- $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \neg\varphi$ iff $\mathbb{A}, \rho \nVdash_{\mathsf{loc}} \varphi$, and
- $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_1 \wedge \varphi_2$ iff $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_1$ and $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_2$.

Definition 2.1.3. An F-*structure* is a pair $(\mathbb{A}, \mathsf{S})$ where $\mathbb{A}$ is an F-algebra with carrier set A and $\mathsf{S} \subseteq A^N$ is a non-empty set of *possible outcomes*.

Definition 2.1.4. Given an F-structure $(\mathbb{A}, \mathsf{S})$, the *satisfaction of global formulas* by an F-structure is defined inductively by:

- $(\mathbb{A}, \mathsf{S}) \Vdash \forall\varphi$ iff $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi$ for every $\rho \in \mathsf{S}$,
- $(\mathbb{A}, \mathsf{S}) \Vdash \neg\delta$ iff $(\mathbb{A}, \mathsf{S}) \nVdash \delta$, and
- $(\mathbb{A}, \mathsf{S}) \Vdash \delta_1 \wedge \delta_2$ iff $(\mathbb{A}, \mathsf{S}) \Vdash \delta_1$ and $(\mathbb{A}, \mathsf{S}) \Vdash \delta_2$.

As usual, given $\Delta \subseteq \mathsf{Glob}$ we write $(\mathbb{A}, \mathsf{S}) \Vdash \Delta$ if $(\mathbb{A}, \mathsf{S}) \Vdash \delta$ for every $\delta \in \Delta$.

Definition 2.1.5. *Semantic consequence* is defined, as usual, by $\Delta \vDash_{\mathcal{A}} \delta$ whenever $(\mathbb{A}, \mathsf{S}) \Vdash \Delta$ implies $(\mathbb{A}, \mathsf{S}) \Vdash \delta$, for any F-structure $(\mathbb{A}, \mathsf{S})$ with $\mathbb{A} \in \mathcal{A}$.

[1] This terminology stems from the intuition that names could be sampled from a distribution. Our aim is, indeed, to add a probabilistic component to this logic, that will materialize in Chapter 4. For the moment, however, outcomes should just be understood as being obtained non-deterministically.

Example 2.1.6. Consider the signature F^{com} where we require $s \in \mathrm{F}_2^{com}$, and let $\mathcal{A}^{com}$ be the class of F^{com}-algebras satisfying the set of equations $\Gamma^{com} = \{s(x_1,x_2) \approx s(x_2,x_1)\}$, i.e., $\mathcal{A}^{com}$ is the class of all commutative groupoids. Then, for $n,m,a,b,c \in N$, we have:

$$\forall(n \approx a \vee n \approx b), \forall(m \approx a \vee m \approx b), \forall(s(a,b) \approx c) \vDash_{\mathcal{A}^{com}} \forall(n \not\approx m \to s(n,m) \approx c),$$

asserting that in a commutative groupoid where the sum of a and b leads to c, the sum of m and n also leads to c provided that one is equal to a and the other takes the value of b. $\triangle$

Example 2.1.7. A standard example of an equational theory used in information security for formalizing (part of) the capabilities of a so-called *Dolev-Yao attacker* (see, for instance, [2,3,18]) consists in taking a signature F^{DY} with $\{\cdot\}_\cdot, \{\cdot\}_\cdot^{-1} \in \mathrm{F}_2^{\mathsf{DY}}$, representing symmetric encryption and decryption of a message with a key, $(\cdot,\cdot) \in \mathrm{F}_2^{\mathsf{DY}}$, representing message pairing, and $\pi_1, \pi_2 \in \mathrm{F}_1^{\mathsf{DY}}$ representing projections. The algebraic properties of these operations are given by

$$\Gamma^{\mathsf{DY}} = \{\{\{x_1\}_{x_2}\}_{x_2}^{-1} \approx x_1, \pi_1(x_1,x_2) \approx x_1, \pi_2(x_1,x_2) \approx x_2\}.$$

Let $\mathcal{A}^{\mathsf{DY}}$ be a Γ^{DY}-variety, i.e., the class of algebras satisfying Γ^{DY}. Then, we have that

$$\vDash_{\mathcal{A}^{\mathsf{DY}}} \forall(m \approx k) \to \forall\left(\{\{n\}_k\}_m^{-1} \approx \pi_2(a,n)\right).$$

$\triangle$

2.2 Deductive System

In order to obtain a sound and complete deductive system for EqCL, we must additionally require that the basic class $\mathcal{A}$ of algebras be axiomatized by a set Γ of Horn clauses over X. From there, we can define the deductive system $\mathcal{H}_\Gamma$ presented in Figure 2.1.

Eq1	$\forall(t \approx t)$	**N1**	$\forall(\varphi_1 \wedge \varphi_2) \leftrightarrow (\forall\varphi_1 \wedge \forall\varphi_2)$
Eq2	$\forall(t_1 \approx t_2 \to t_2 \approx t_1)$	**N2**	$\forall\neg\varphi \to \neg\forall\varphi$
Eq3	$\forall(t_1 \approx t_2 \ \wedge \ t_2 \approx t_3 \to t_1 \approx t_3)$	**N3**	$\neg\forall\varphi \to \forall\neg\varphi$, if $\mathsf{names}(\varphi) = \varnothing$
Eq4	$\forall(t_1 \approx t_1' \wedge \ldots \wedge t_n \approx t_n' \to f(t_1,\ldots,t_n) \approx f(t_1',\ldots,t_n'))$	**N4**	$\forall(\varphi_1 \leftrightarrow \varphi_2) \to (\forall\varphi_1 \leftrightarrow \forall\varphi_2)$
EqC1	$\forall((\varphi_1 \to (\varphi_2 \to \varphi_3)) \to ((\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)))$	**C1**	$\delta_1 \to (\delta_2 \to \delta_1)$
EqC2	$\forall(\varphi_1 \to (\varphi_2 \to \varphi_1))$	**C2**	$(\delta_1 \to (\delta_2 \to \delta_3)) \to ((\delta_1 \to \delta_2) \to (\delta_1 \to \delta_3))$
EqC3	$\forall((\neg\varphi_1 \to \neg\varphi_2) \to (\varphi_2 \to \varphi_1))$	**C3**	$(\neg\delta_1 \to \neg\delta_2) \to (\delta_2 \to \delta_1)$
EqC4	$\forall(\varphi_1 \to ((\varphi_1 \to \varphi_2) \to \varphi_2))$	**C4**	$\frac{\delta_1 \quad \delta_1 \to \delta_2}{\delta_2}$
E(Γ)	$\forall((\sigma(s_1) \approx \sigma(s_1') \wedge \ldots \wedge \sigma(s_n) \approx \sigma(s_n')) \to \sigma(s) \approx \sigma(s'))$		

for every $t, t_1, t_2, t_3, \ldots, t_n, t_1', \ldots, t_n' \in T(N)$, $\varphi, \varphi_1, \varphi_2, \varphi_3 \in \mathsf{Loc}$, $\delta_1, \delta_2, \delta_3 \in \mathsf{Glob}$, $\sigma \in T(N)^X$ and $(s_1 \approx s_1', \ldots, s_n \approx s_n' \Rightarrow s \approx s') \in \Gamma$.

Figure 2.1: The deductive system $\mathcal{H}_\Gamma$.

The deductive system $\mathcal{H}_\Gamma$ consists of a number of axioms and inference rules $\mathsf{C4}$, *modus ponens*. The system combines the different components inherent to this logic: axioms $\mathsf{Eq1}$-$\mathsf{Eq4}$ incorporate standard equational reasoning, namely reflexivity, symmetry, transitivity and congruence; $\mathsf{C1}$-$\mathsf{C4}$ and $\mathsf{EqC1}$-$\mathsf{EqC4}$ incorporate classical reasoning for the global and local layers (just note that locally, *modus ponens* becomes axiom $\mathsf{EqC4}$); $\mathsf{N1}$-$\mathsf{N4}$ characterize the relationship between the local and global layers across the universal quantifier; and the axioms $\mathsf{E}(\Gamma)$ incorporate the equational theory underlying $\mathcal{A}$. We define, as usual, a deducibility relation $\vdash_\Gamma^{\mathrm{F}}$. We drop the superscript F whenever it is clear from context.

EqCL is an extension of classical logic at both the local and the global layers. Hence, we are able to import many properties and results from classical propositional logic with similar proofs, just by noting that the inference rule $\mathsf{C4}$ is *modus ponens*. Namely, it is easy to see that the *deduction meta-theorem* holds, that it is possible to rewrite any global formula into disjunctive normal form, and so on. The following Lemma is the compilation some of these results, that will be useful later.

Lemma 2.2.1. *The following properties hold:*

MTD $\Psi \cup \{\delta\} \vdash_\Gamma \delta'$ *if and only if* $\Psi \vdash_\Gamma \delta \to \delta'$.

Aux1 $\vdash_\Gamma \forall((\varphi_1 \to \varphi_2) \leftrightarrow \neg(\varphi_1 \wedge \neg\varphi_2))$

Aux2 $\vdash_\Gamma \delta_1 \to (\delta_2 \to (\delta_1 \wedge \delta_2))$

Aux3 $\vdash_\Gamma \forall((\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1) \leftrightarrow \varphi_1 \wedge \varphi_2)$

Aux4 $\vdash_\Gamma (\forall\varphi_1 \wedge \forall\varphi_2) \to \forall\varphi_2$

DNF $\vdash_\Gamma \delta \leftrightarrow \bigvee_{j=1}^{m}\left(\bigwedge_{i=1}^{n_j} \delta_i^j\right)$, *for some* $\{\delta_i^j\}_{\substack{i\in\{1,\dots,n_j\}\\ j\in\{1,\dots,m\}}} \subseteq (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc})$,

where $\varphi_1, \varphi_2, \varphi_3 \in \mathsf{Loc}$, $\Psi \subseteq \mathsf{Glob}$, $\delta, \delta', \delta_1, \delta_2, \delta_3 \in \mathsf{Glob}$.

Proof. The proof of each of these properties follows by a simple replication of what is done for its analogue in classical propositional logic or either by simple observation of the introduced abbreviations for the connectives. For the sake of illustration, we sketch the proof of MTD and DNF.

MTD To prove the reverse implication, consider a deduction $\delta_1, \dots, \delta_r$ of $\delta \to \delta'$ from Ψ, where the formula δ_r is $\delta \to \delta'$, and let us use that to construct the following deduction of δ'

from $\Psi \cup \{\delta\}$:

$$\begin{array}{rll}
\mathsf{s}_1. & \delta_1 & \\
 & \vdots & \\
\mathsf{s}_\mathsf{r}. & \delta_r & \text{(deduction of } \delta \to \delta'\text{)} \\
\mathsf{s}_{\mathsf{r+1}}. & \delta & (\delta \in \Psi \cup \{\delta\}) \\
\mathsf{s}_{\mathsf{r+2}}. & \delta' & \text{(from } \mathsf{s}_\mathsf{r} \text{ and } \mathsf{s}_{\mathsf{r+1}} \text{ by } \mathsf{C4}\text{)}
\end{array}$$

For the direct implication, let us consider a deduction $\gamma_1, \dots, \gamma_\ell$ of δ' from $\Psi \cup \{\delta\}$ and let us prove that $\Psi \vdash_\Gamma \delta \to \delta'$ by induction on the length of the deduction, ℓ. Recall that γ_ℓ is δ' and notice that for $\ell = 1$ the deduction of δ' from $\Psi \cup \{\delta\}$ is simply

$$\begin{array}{rl}
\mathsf{s}_1. & \delta'
\end{array}$$

where either δ' is an instance of an axiom, $\delta' \in \Psi$ or $\delta' \in \{\delta\}$. For each of these three cases we construct a deduction of $\delta \to \delta'$ from Ψ.

1. If δ' is an instance of an axiom, we have the following deduction of $\Psi \vdash_\Gamma \delta \to \delta'$:

$$\begin{array}{rll}
\mathsf{s}_1. & \delta' & \text{(instance of an axiom of } \mathcal{H}_\Gamma\text{)} \\
\mathsf{s}_2. & \delta' \to (\delta \to \delta') & \text{(instance of } \mathsf{C1}\text{)} \\
\mathsf{s}_3. & \delta \to \delta' & \text{(from } \mathsf{s}_1 \text{ and } \mathsf{s}_2 \text{ by } \mathsf{C4}\text{)}
\end{array}$$

2. If $\delta' \in \Psi$, we conclude that $\Psi \vdash_\Gamma \delta \to \delta'$ using the following deduction:

$$\begin{array}{rll}
\mathsf{s}_1. & \delta' & (\delta' \in \Psi) \\
\mathsf{s}_2. & \delta' \to (\delta \to \delta') & \text{(instance of } \mathsf{C1}\text{)} \\
\mathsf{s}_3. & \delta \to \delta' & \text{(from } \mathsf{s}_1 \text{ and } \mathsf{s}_2 \text{ by } \mathsf{C4}\text{)}
\end{array}$$

3. If $\delta' \in \{\delta\}$, i.e. δ' coincides with δ, consider the following deduction of $\delta \to \delta$:

$$\begin{array}{rll}
\mathsf{s}_1. & (\delta \to ((\delta \to \delta) \to \delta)) \to ((\delta \to (\delta \to \delta)) \to (\delta \to \delta)) & \text{(instance of } \mathsf{C2}\text{)} \\
\mathsf{s}_2. & \delta \to ((\delta \to \delta) \to \delta) & \text{(instance of } \mathsf{C1}\text{)} \\
\mathsf{s}_3. & (\delta \to (\delta \to \delta)) \to (\delta \to \delta) & \text{(from } \mathsf{s}_1 \text{ and } \mathsf{s}_2 \text{ by } \mathsf{C4}\text{)} \\
\mathsf{s}_4. & \delta \to (\delta \to \delta) & \text{(instance of } \mathsf{C1}\text{)} \\
\mathsf{s}_5. & \delta \to \delta & \text{(from } \mathsf{s}_3 \text{ and } \mathsf{s}_4 \text{ by } \mathsf{C4}\text{)}
\end{array}$$

   i.e., $\vdash_\Gamma \delta \to \delta$, which implies $\Psi \vdash_\Gamma \delta \to \delta$.

Now assume that whenever the length of the deduction of $\Psi \cup \{\delta\} \vdash_\Gamma \delta'$ is ℓ, there exists a deduction of $\delta \to \delta'$ from Ψ and let us prove the same for deductions of length $\ell + 1$. Assume that there exists a deduction of $\Psi \cup \{\delta\} \vdash_\Gamma \delta'$ with length ℓ+1: $\gamma_1, \dots, \gamma_{\ell+1}$. Recall that $\gamma_{\ell+1}$ coincides with δ' notice that: if either the $(\ell+1)^{th}$ formula of the deduction, δ', is an instance of an axiom, $\delta' \in \Psi$ or $\delta' \in \{\delta\}$, then we proceed as in the base case; otherwise, $\gamma_{\ell+1}$ should follow from inference rule $\mathsf{C4}$ applied to two previous formulas in the deduction, say ψ and $\psi \to \delta'$. Since both ψ and $\psi \to \delta'$ appear in the deduction before the $(\ell+1)^{th}$ line, by induction hypothesis there are proofs of $\delta \to (\psi \to \delta')$ from Ψ, say $\psi_1, \ldots, \psi_n$, and of $\delta \to \psi$ from Ψ, call it $\psi'_1, \ldots, \psi'_m$, so that consider the following deduction of $\delta \to \delta'$ from Ψ:

$\mathsf{s_1}$. ψ_1

$\vdots$

$\mathsf{s_n}$. ψ_n (deduction of $\varphi \to (\psi \to \varphi')$ from Ψ)

$\mathsf{s_{n+1}}$. ψ'_1

$\vdots$

$\mathsf{s_{n+m}}$. ψ'_m (deduction of $\varphi \to \psi$ from Ψ)

$\mathsf{s_{n+m+1}}$. $(\delta \to (\psi \to \delta')) \to ((\delta \to \psi) \to (\delta \to \delta'))$ (instance of $\mathsf{C2}$)

$\mathsf{s_{n+m+2}}$. $(\delta \to \psi) \to (\delta \to \delta')$ (from $\mathsf{s_n}$ and $\mathsf{s_{n+m+1}}$ by $\mathsf{C4}$)

$\mathsf{s_{n+m+3}}$. $\delta \to \delta'$ (from $\mathsf{s_{n+m}}$ and $\mathsf{s_{n+m+2}}$ by $\mathsf{C4}$)

We proved that $\Psi \vdash_\Gamma \delta \to \delta'$.

DNF Let $\delta \in \mathsf{Glob}$ be any formula. We prove by structural induction over the formula δ that it can be converted into DNF.

For the base case, if δ is of the form $\forall\varphi$ for some $\varphi \in \mathsf{Loc}$, it is already in the DNF. So, let us assume that all subformulas of δ can be written in DNF.

- If δ is of the form $\neg\delta'$ for $\delta' \in \mathsf{Glob}$, by induction hypothesis, δ' can be written in DNF:

$$\delta' \text{ is equivalent to } \bigvee_{j=1}^{n}\left(\bigwedge_{i=1}^{m_j}\delta_i^j\right) \text{ for some } \{\delta_i^j\}_{\substack{i\in\{1,\ldots,m_j\}\\ j\in\{1,\ldots,n\}}} \subseteq (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}).$$

  But then, making use of the introduced abbreviations for disjunction, we can write $\neg\delta'$ equivalently as $\neg\left(\bigvee_{j=1}^{n}\left(\bigwedge_{i=1}^{m_j}\delta_i^j\right)\right) = \bigwedge_{j=1}^{n}\neg\left(\bigwedge_{i=1}^{m_j}\delta_i^j\right) = \bigwedge_{j=1}^{n}\left(\bigvee_{i=1}^{m_j}\neg\delta_i^j\right)$. Notice that $\{\neg\delta_i^j\}_{\substack{i\in\{1,\ldots,m_j\}\\ j\in\{1,\ldots,n\}}} \subseteq (\forall\mathsf{Loc}\cup\neg\forall\mathsf{Loc})$. By the distributivity of conjunction over disjunction, it can be written equivalently as $\bigvee_{j=1}^{s}\left(\bigwedge_{i=1}^{r_j}\psi_i^j\right)$, for some positive integers $s, r_1, \ldots, r_s$ and literals $\{\psi_i^j\}_{\substack{i\in\{1,\ldots,r_j\}\\ j\in\{1,\ldots,s\}}} \subseteq (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc})$.
- If δ is of the form $\delta_1 \wedge \delta_2$ for some $\delta_1, \delta_2 \in \mathsf{Glob}$ then, by induction hypothesis, δ_1 and δ_2 can both be written in the DNF:

$$\delta_1 \text{ is equivalent to } \bigvee_{j=1}^{n_1}\left(\bigwedge_{i=1}^{m_1^j}\delta_i^j\right) \text{ and } \delta_2 \text{ is equivalent to } \bigvee_{j=1}^{n_2}\left(\bigwedge_{i=1}^{m_2^j}\bar{\delta}_i^j\right).$$

But then, $\delta_1 \wedge \delta_2$ is equivalent to $\left(\bigvee_{j=1}^{n_1}\left(\bigwedge_{i=1}^{m_1^j} \delta_i^j\right)\right) \wedge \left(\bigvee_{j=1}^{n_2}\left(\bigwedge_{i=1}^{m_2^j} \bar{\delta}_i^j\right)\right)$. Applying the distributivity of conjunction over disjunction $\delta_1 \wedge \delta_2$ is still equivalent to $\bigvee_{j=1}^{n}\left(\bigwedge_{i=1}^{m_j} \psi_i^j\right)$, for some positive integers $n, m_1, \ldots, m_n$ and literals $\{\psi_i^j\}_{\substack{i \in \{1,\ldots,m_j\} \\ j \in \{1,\ldots,n\}}} \subseteq (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc})$.

□

We should observe that the semantics assigned to the global formulas in $\forall\mathsf{Loc}$ supports the feeling that this quantification can be understood as an $S5$-like modality. Notice that a normality-like axiom can be easily derived.

Lemma 2.2.2. *Given* $\varphi_1, \varphi_2 \in \mathsf{Loc}$, $\vdash_\Gamma \forall(\varphi_1 \to \varphi_2) \to (\forall\varphi_1 \to \forall\varphi_2)$.

Proof. To deduce $\vdash_\Gamma \forall(\varphi_1 \to \varphi_2) \to (\forall\varphi_1 \to \forall\varphi_2)$, we employ the deduction meta-theorem MTD and construct a deduction of $\forall\varphi_2$ from $\{\forall(\varphi_1 \to \varphi_2), \forall\varphi_1\}$ to prove that

$$\{\forall(\varphi_1 \to \varphi_2), \forall\varphi_1\} \vdash_\Gamma \forall\varphi_2.$$

s_1. $\forall(\varphi_1 \to \varphi_2)$ (hypothesis)

s_2. $\forall\left((\varphi_1 \to \varphi_2) \leftrightarrow \neg(\varphi_1 \wedge \neg\varphi_2)\right) \to \left(\forall(\varphi_1 \to \varphi_2) \leftrightarrow \forall\neg(\varphi_1 \wedge \neg\varphi_2)\right)$ (instance of N4)

s_3. $\forall\left((\varphi_1 \to \varphi_2) \leftrightarrow \neg(\varphi_1 \wedge \neg\varphi_2)\right)$ (instance of Aux1)

s_4. $\forall(\varphi_1 \to \varphi_2) \leftrightarrow \forall\neg(\varphi_1 \wedge \neg\varphi_2)$ (apply C4 to s_2 and s_3)

s_5. $\forall\neg(\varphi_1 \wedge \neg\varphi_2)$ (apply C4 to s_1 and s_4)

s_6. $\forall\varphi_1$ (hypothesis)

s_7. $\forall\neg(\varphi_1 \wedge \neg\varphi_2) \to (\forall\varphi_1 \to (\forall\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \forall\varphi_1))$ (instance of Aux2)

s_8. $\forall\varphi_1 \to (\forall\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \forall\varphi_1)$ (apply C4 to s_5 and s_7)

s_9. $\forall\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \forall\varphi_1$ (apply C4 to s_6 and s_8)

s_{10}. $\forall\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \forall\varphi_1 \leftrightarrow \forall\left(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1\right)$ (instance of N1)

s_{11}. $\forall\left(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1\right)$ (apply C4 to s_9 and s_{10})

s_{12}. $\forall\left(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1 \leftrightarrow \varphi_1 \wedge \varphi_2\right)$ (instance of Aux3)

s_{13}. $\forall\left(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1 \leftrightarrow \varphi_1 \wedge \varphi_2\right) \to$
$\to \left(\forall(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1) \leftrightarrow \forall(\varphi_1 \wedge \varphi_2)\right)$ (instance of N4)

s_{14}. $\forall\left(\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1\right) \leftrightarrow \forall(\varphi_1 \wedge \varphi_2)$ (apply C4 to s_{12} and s_{13})

s_{15}. $\forall(\varphi_1 \wedge \varphi_2)$ (apply C4 to s_{11} and s_{14})

s_{16}. $\forall(\varphi_1 \wedge \varphi_2) \leftrightarrow \forall\varphi_1 \wedge \forall\varphi_2$ (instance of N1)

s_{17}. $\forall\varphi_1 \wedge \forall\varphi_2$ (apply C4 to s_{15} and s_{16})

s_{18}. $\forall\varphi_1 \wedge \forall\varphi_2 \to \forall\varphi_2$ (instance of Aux4)

s_{19}. $\forall\varphi_2$ (apply C4 to s_{17} and s_{18})

□

Example 2.2.3. Recall Example 2.1.6. From the commutativity equation we obtain the axiom $\forall(s(n_1,n_2) \approx s(n_2,n_1))$, for $n_1, n_2 \in N$. By using also Eq3-Eq4, EqC1-EqC4, N1, and finally applying inference rule C4, we can easily show that

$$\forall(n \approx a \vee n \approx b), \forall(m \approx a \vee m \approx b), \forall(s(a,b) \approx c) \vdash_{\Gamma^{com}} \forall(n \not\approx m \to s(n,m) \approx c). \qquad \triangle$$

2.3 Soundness and Completeness

We now prove that $\mathcal{H}_\Gamma$ is a sound and complete deductive system for the logic based on the class $\mathcal{A}$ of all algebras that satisfy Γ. In this way, we ensure that the syntactic consequences derived from $\mathcal{H}_\Gamma$ correspond exactly to the conclusions that we can entail from the semantics that we have set out.

Theorem 2.3.1. *The deductive system $\mathcal{H}_\Gamma$ is sound.*

Proof. The proof of soundness follows by induction on the structure of the deductions. In this sense, we must prove that each axiom and the inference rule of $\mathcal{H}_\Gamma$ are valid. The proof of validity of all axioms and inference rules is straightforward, however we detail some of them by their different nature.

Eq4 To prove that Eq4 is valid, consider an F-structure $(\mathbb{A}, \mathsf{S})$ with $\mathbb{A} \in \mathcal{A}$ and let $\rho \in \mathsf{S}$ represent any possible outcome for which $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} t_i \approx t'_i$, i.e, $[\![t_i]\!]^\rho_\mathbb{A} = [\![t'_i]\!]^\rho_\mathbb{A}$ for each $i \in \{1,\ldots,n\}$. Then, obviously $[\![f(t_1,...,t_n)]\!]^\rho_\mathbb{A} = [\![f(t'_1,...,t'_n)]\!]^\rho_\mathbb{A}$, and $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} f(t_1,...,t_n) \approx f(t'_1,...,t'_n)$.

N2 For N2, let $\varphi \in \mathsf{Loc}$ be any local formula and let $(\mathbb{A}, \mathsf{S})$ be an F-structure with $\mathbb{A} \in \mathcal{A}$ such that $(\mathbb{A}, \mathsf{S}) \Vdash \forall\neg\varphi$, i.e., $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \neg\varphi$ for every $\rho \in \mathsf{S}$. It means that there exists at least one $\rho \in \mathsf{S}$ for which $\mathbb{A}, \rho \nVdash_{\mathsf{loc}} \varphi$ and we easily conclude that $(\mathbb{A}, \mathsf{S}) \Vdash \neg\forall\varphi$.

N3 To check that N3 is valid, let $\varphi \in \mathsf{Loc}$ be a local formula such that $\mathsf{names}(\varphi) = \varnothing$ and consider $(\mathbb{A}, \mathsf{S})$ to be any F-structure with $\mathbb{A} \in \mathcal{A}$ and such that $(\mathbb{A}, \mathsf{S}) \Vdash \neg\forall\varphi$, i.e., there exists an outcome $\rho^* \in \mathsf{S}$ such that $\mathbb{A}, \rho^* \Vdash_{\mathsf{loc}} \neg\varphi$. Since $\mathsf{names}(\varphi) = \varnothing$, the satisfiability of φ does not depend on the outcome $\rho \in \mathsf{S}$, so that $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \neg\varphi$ for every $\rho \in \mathsf{S}$. Hence, $(\mathbb{A}, \mathsf{S}) \Vdash \forall\neg\varphi$.

E(Γ) For the validity of $\mathsf{E}(\Gamma)$, consider a Horn clause $s_1 \approx s'_1, \ldots, s_n \approx s'_n \Rightarrow s \approx s' \in \Gamma$, a substitution $\sigma \in T(N)^X$ and an F-structure $(\mathbb{A}, \mathsf{S})$ with $\mathbb{A} \in \mathcal{A}$ and let us check that $(\mathbb{A}, \mathsf{S}) \Vdash \forall((\sigma(s_1) \approx \sigma(s'_1) \wedge \ldots \wedge \sigma(s_n) \approx \sigma(s'_n)) \to \sigma(s) \approx \sigma(s'))$. For this purpose, consider $\rho : N \to A$ to be any outcome in S and assume that $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \sigma(s_1) \approx \sigma(s'_1) \wedge \ldots \wedge \sigma(s_n) \approx \sigma(s'_n)$. By noting that $[\![\cdot]\!]^\rho_\mathbb{A} \circ \sigma : X \to A$ and recalling that $\mathbb{A} \in \mathcal{A}$, i.e., the algebra $\mathbb{A}$ satisfies all clauses in Γ, we easily conclude that $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \sigma(s) \approx \sigma(s')$. □

Remark 2.3.2. Recall that a set of global formulas $\Delta \subseteq \mathsf{Glob}$ is consistent if there exists $\delta \in \mathsf{Glob}$ such that $\Delta \nvdash_{\Gamma} \delta$. Since the EqCL is classically based, $\Delta \nvdash_{\Gamma} \delta$ if and only if $\Delta \cup \{\neg\delta\}$ is consistent. Furthermore, as a consequence of Lindenbaum's Lemma and given any set K, we have that $\{\bigvee_{i=1}^{n_k} \delta_{k,i} \mid k \in K\}$ is consistent if and only if, for every $k \in K$, there exists $1 \leq i_k \leq n_k$ such that $\{\delta_{k,i_k} \mid k \in K\}$ is consistent.

Theorem 2.3.3. *The deductive system $\mathcal{H}_{\Gamma}$ is complete.*

The proof of completeness that we will present after a few technical details, is the combination of several well-know techniques used to deal with equalities, quantifiers and classical reasoning. We follow the usual contrapositive approach and prove that: given $\Delta \subseteq \mathsf{Glob}$ and $\delta \in \mathsf{Glob}$, if $\Delta \nvdash_{\Gamma} \delta$ then $\Delta \nvDash_{\mathcal{A}} \delta$. Let us fix the context and prove some auxiliary results that will be useful throughout the proof. Once we assume that $\Delta \nvdash_{\Gamma} \delta$, we will be searching for an F-structure $(\mathbb{A}, \mathsf{S})$ such that $\mathbb{A}$ satisfies Γ, $(\mathbb{A}, \mathsf{S}) \Vdash \Delta$ and $(\mathbb{A}, \mathsf{S}) \nVdash \delta$. With this mindset, let us begin by writing each element of $\Delta \cup \{\neg\delta\}$ in disjunctive normal form:

$$\left\{ \xi^{\mathsf{DNF}} := \bigvee_{j=1}^{m_\xi} \bigwedge_{i=1}^{n_j} \psi_{\xi,j,i} \mid \xi \in \Delta \cup \{\neg\delta\} \right\}, \tag{2.1}$$

where $m_\xi, n_j \in \mathbb{N}$, and $\psi_{\xi,j,i} \in (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc})$. Then, let

$$\left\{ \bigwedge_{i=1}^{n_{j_\xi}} \psi_{\xi,j_\xi,i} \mid \xi \in \Delta \cup \{\neg\delta\} \right\} \quad \text{be a consistent set} \tag{2.2}$$

constructed by one disjunct of each element in (2.1), according to Remark 2.3.2.

We will be looking for an F-structure satisfying each of the following *relevant atoms*:

$$\mathsf{RelAt}(\Delta \cup \{\neg\delta\}) = \bigcup_{\xi \in \Delta \cup \{\neg\delta\}} \left\{ \psi_{\xi,j_\xi,1}, \ldots, \psi_{\xi,j_\xi,n_{j_\xi}} \right\} \subseteq \forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}\,. \tag{2.3}$$

We follow a Henkin construction [69] to define the F-algebra $\mathbb{A}$, adding enough constants to the language in order to introduce all the necessary witnesses for formulas of the form $\neg\forall\varphi$. Note that the set of local formulas is countable and thus we introduce a set of new constants for each of them, which we will use to instantiate all names in N: $\bigcup_{\varphi \in \mathsf{Loc}} \{c_{\varphi,n} \mid n \in N\}$. The extended signature by F^+ coincides with F in all but

$$\mathrm{F}_0^+ = \mathrm{F}_0 \cup \left(\bigcup_{\varphi \in \mathsf{Loc}} \{c_{\varphi,n} \mid n \in N\} \right).$$

The set (2.3) is extended with the witnesses for existential formulas by considering an enumeration for $\mathsf{Loc} \times \mathsf{Loc}$ and then considering the following inductive definition:

$$\begin{aligned} W_0 &= \mathsf{RelAt}(\Delta \cup \{\neg\delta\}) \\ W_{i+1} &= W_i \cup \left\{ \neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \right) \right) \right\} \quad \text{for each } i \in \mathbb{N}, \end{aligned}$$

where $\mathsf{names}(\varphi_i^1) \cup \mathsf{names}(\varphi_i^2) = \tilde{n} = \{n_1, \dots, n_m\}$, $\tilde{c}_\varphi = \{c_{\varphi,n_1}, \dots, c_{\varphi,n_m}\}$. This way, given $i \in \mathbb{N}$ we introduce, where appropriate, a witness for $\neg\forall\varphi_i^1$.

To prove that the union of this family of sets is consistent with respect to the extended signature F^+, we need to prove an auxiliary Lemma.

Lemma 2.3.4. *If* $W_i \vdash_\Gamma^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)$ *then*

$$W_i \vdash_\Gamma^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^1\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall\varphi_i^2\right).$$

Proof. Assume that $W_i \vdash_\Gamma^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)$ and let $\omega = \psi_1 ... \psi_p$ denote a deduction of it. Note that we can change all the new constants $\tilde{c}_{\varphi_i^1}$ back to names, $[\omega]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} = [\psi_1]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} \dots [\psi_n]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$, and obtain a deduction of

$$[W_i]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} \vdash_\Gamma^{\mathrm{F}^+} \left[\left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} \quad .$$

Let us check this. Given $l \in \{1, \dots, p\}$,

- if $\psi_\ell \in W_i$, $[\psi_\ell]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} \in [W_i]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$,
- if $\psi_\ell \in T_{\mathrm{F}^+}(N)$ is an instance of an axiom, $[\psi_j]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$ is an instance of the same axiom,
- if ψ_ℓ results from applying $\mathsf{C4}$ to ψ_{l_1}, ψ_{l_2}, where ψ_{l_2} should be of the form $\psi_{l_1} \to \psi_\ell$, then $[\psi_\ell]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$ results from applying $\mathsf{C4}$ to $[\psi_{l_1}]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}, [\psi_{l_2}]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$.

We proved by induction in the structure of the deduction ω that

$$[W_i]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} \vdash_\Gamma^{\mathrm{F}^+} \left[\left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} .$$

According to definition of replacement for global formulas and since $\tilde{c}_{\varphi_i^1}$ are new constants occurring nowhere else, $\left[\left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}}$ is the following global formula $\forall\left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^1\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall\varphi_i^2\right)$. On the other hand, $\tilde{c}_{\varphi_i^1}$ does not occur in W_i, and so, $[W_i]_{\tilde{n}}^{\tilde{c}_{\varphi_i^1}} = W_i$. This shows that $W_i \vdash_\Gamma^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^1\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall\varphi_i^2\right)$. □

We are now able to prove the main Lemma, which states that the limit of the previous nondecreasing sequence of sets $\{W_i\}_{i \in \mathbb{N}}$ is consistent.

Lemma 2.3.5. $W = \bigcup_{i \in \mathbb{N}} W_i$ *is consistent (regarding* F^+*).*

Proof. Let us prove by induction on $i \in \mathbb{N}$ that W_i is consistent: $W_0 = \mathsf{RelAt}(\Delta \cup \{\neg\delta\})$ is consistent immediately from (2.2). Then assume that W_i is consistent but W_{i+1} is not consistent. In this case, $W_{i+1} \vdash_{\Gamma}^{\mathrm{F}^+} \delta$ for any $\delta \in \mathsf{Glob}_{\mathrm{F}^+}$. In particular,

$$W_{i+1} \vdash_{\Gamma}^{\mathrm{F}^+} \neg\left(\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right).$$

Recalling that $W_{i+1} = W_i \cup \left\{\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right\}$ and using the deduction meta-theorem MTD it follows that

$$W_i \vdash_{\Gamma}^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right) \to \left(\neg\left(\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right)\right),$$

i.e., $W_i \vdash_{\Gamma}^{\mathrm{F}^+} \neg\left(\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right)$.

We can write the following equivalent assertions:

$$\begin{aligned}
& W_i \vdash_{\Gamma}^{\mathrm{F}^+} \neg\left(\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right) \\
\textsf{iff} \quad & W_i \vdash_{\Gamma}^{\mathrm{F}^+} \neg\forall\varphi_i^1 \wedge \neg\left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right) \\
\textsf{iff} \quad & W_i \vdash_{\Gamma}^{\mathrm{F}^+} \neg\forall\varphi_i^1 \wedge \left(\forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \vee \neg\left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right) \\
\textsf{iff} \quad & W_i \vdash_{\Gamma}^{\mathrm{F}^+} \neg\forall\varphi_i^1 \wedge \left(\forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \vee \left(\forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right) \\
\textsf{iff} \quad & W_i \vdash_{\Gamma}^{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall[\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right),
\end{aligned}$$

which, using Lemma 2.3.4, leads to

$$W_i \vdash_{\mathrm{F}^+} \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^1\right) \vee \left(\neg\forall\varphi_i^1 \wedge \forall\varphi_i^2 \wedge \neg\forall\varphi_i^2\right), \tag{2.4}$$

which is a contradiction and ends our proof that W is consistent with respect to F^+. □

Now that we have some ingredients, we proceed with the proof of completeness.

Proof of Theorem 2.3.3. Consider $\Delta \subseteq \mathsf{Glob}$, $\delta \in \mathsf{Glob}$ and assume $\Delta \nvdash_{\Gamma} \delta$. We need to prove that $\Delta \nvDash_{\mathcal{A}} \delta$ by defining an F-structure $(\mathbb{A}, \mathsf{S})$ such that $\mathbb{A}$ satisfies Γ, $(\mathbb{A}, \mathsf{S}) \Vdash \Delta$ and $(\mathbb{A}, \mathsf{S}) \nVdash \delta$, i.e., $(\mathbb{A}, \mathsf{S})$ should satisfy the set of relevant atoms in (2.3), $(\mathbb{A}, \mathsf{S}) \Vdash \mathsf{RelAt}(\Delta \cup \{\neg\delta\})$.

To define such F-structure, recall the Henkin construction carried out previously and consider the set $W = \bigcup_{i\in\mathbb{N}} W_i \subseteq \mathsf{Glob}_{\mathrm{F}^+}$ that was proved to be consistent in Lemma 2.3.5, regarding the extended signature F^+. Let $\Xi \subseteq \mathsf{Glob}_{\mathrm{F}^+}$ be a maximal consistent set extending W, whose existence is guaranteed by the Lindenbaum's Lemma (Lemma 1.1.8), and consider the congruence relation $\equiv$ over $T_{\mathrm{F}^+}(N)$ defined by $t_1 \equiv t_2$ iff $\forall(t_1 \approx t_2) \in \Xi$. Axioms $\mathsf{Eq1}$-$\mathsf{Eq4}$ together with Lemma 2.2.2, make $\equiv$ be a congruence relation. We define $\mathbb{A}$ to be the

the quotient F^+-algebra $\mathbb{T}_{\mathrm{F}^+}(N)_{/\equiv}$. Note that by definition of $\equiv$, by $\mathsf{E}(\Gamma)$, $\mathsf{C4}$, Lemma 2.2.2, and recalling that Ξ is maximally consistent, it is easy to check that $\mathbb{A}$ satisfies Γ. For the construction of S we choose to define an outcome for each element of $\neg\forall\mathsf{Loc}$ in Ξ. Given $\neg\forall\varphi \in \Xi$, let $\rho^{\neg\forall\varphi} : N \to A$ be the outcome defined by $\rho(n) = [c_{\varphi,n}]_{\equiv}$ for each $n \in N$. Finally, define $\mathsf{S} = \{\rho^{\neg\forall\varphi} \mid \neg\forall\varphi \in \Xi\}$. Note that $\mathsf{S} \neq \varnothing$ because, given $t \in T(N)$, axiom $\mathsf{Eq1}$ implies that $\forall(\neg(t \not\approx t)) \in \Xi$, which together with axiom $\mathsf{N2}$ means that $\neg\forall(t \not\approx t) \in \Xi$.

In order to conclude that actually $(\mathbb{A}, \mathsf{S}) \Vdash \mathsf{RelAt}(\Delta \cup \{\neg\delta\})$, observe that we can easily prove by induction on the complexity of $\varphi \in \mathsf{Loc}$ that: given $\neg\forall\varphi_0 \in \Xi$,

$$\forall[\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}} \in \Xi \text{ if and only if } \mathbb{A}, \rho^{\neg\forall\varphi_0} \Vdash_{\mathsf{loc}} [\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}}, \text{ where } \mathsf{names}(\varphi) = \tilde{n}. \tag{2.5}$$

To prove that $(\mathbb{A}, \mathsf{S})$ is a model for $\mathsf{RelAt}(\Delta \cup \{\neg\delta\})$, recall that $\mathsf{RelAt}(\Delta \cup \{\neg\delta\}) \subseteq \forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}$, let $\gamma \in \mathsf{RelAt}(\Delta \cup \{\neg\delta\})$ and then prove that $(\mathbb{A}, \mathsf{S}) \Vdash \gamma$ by analyzing the two possible cases for γ:

- if γ is of the form $\forall\varphi$ with $\mathsf{names}(\varphi) = \tilde{n}$, we need to prove that for any $\rho \in \mathsf{S}$ $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi$. Let $\rho \in \mathsf{S}$ and recall that ρ was motivated by some $\neg\forall\varphi_0 \in \Xi$, say that $\rho = \rho^{\neg\forall\varphi_0}$. Since $\forall\varphi \in \mathsf{RelAt}(\Delta \cup \{\neg\delta\}) \subseteq \Xi$ it follows that $\forall[\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}} \in \Xi$ by construction of W. By (2.5) we conclude that $\mathbb{A}, \rho^{\neg\forall\varphi_0} \Vdash_{\mathsf{loc}} [\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}}$, which according to definition of $\rho^{\neg\forall\varphi_0}$ implies that $\mathbb{A}, \rho^{\neg\forall\varphi_0} \Vdash_{\mathsf{loc}} \varphi$.
- on the other hand, if γ is of the form $\neg\forall\varphi$ with $\mathsf{names}(\neg\varphi) = \mathsf{names}(\varphi) = \tilde{n}$, consider the already defined outcome $\rho^{\neg\forall\varphi} \in \mathsf{S}$. Notice that since $\neg\forall\varphi \in \Xi$ it follows that $\forall[\neg\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi}} \in \Xi$. The observation (2.5) implies that $\mathbb{A}, \rho^{\neg\forall\varphi} \Vdash_{\mathsf{loc}} [\neg\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi}}$, which by definition of $\rho^{\neg\forall\varphi}$ implies $\mathbb{A}, \rho^{\neg\forall\varphi} \Vdash_{\mathsf{loc}} \neg\varphi$. Therefore $(\mathbb{A}, \mathsf{S}) \Vdash \neg\forall\varphi$.

We conclude that $(\mathbb{A}, \mathsf{S})$ satisfies the set defined in (2.2), and therefore $(\mathbb{A}, \mathsf{S}) \Vdash \Delta \cup \{\neg\delta\}$. Hence, $\Delta \nvDash_{\mathcal{A}} \delta$. □

This concludes the soundness and completeness results for EqCL. We are now able to use indistinctively the syntactic consequence relation $\vdash_\Gamma$ and the semantic consequence relation $\vDash_{\mathcal{A}}$ provided that the basic class $\mathcal{A}$ of algebras is axiomatized by the set Γ of Horn clauses over X.

The development of other calculi as sequent calculus [63] or labelled tableaux [17, 109] would be interesting and would ease syntactic deductions, however was outside the scope of what we intended to do. Even though, it constitutes an interesting topic for future work.

2.4 Decidability and Complexity

In general, EqCL cannot be expected to be decidable, as equational theories can easily be undecidable [13]. *Combinatory logic* [46], that we present in Example 2.4.1, is an example of an undecidable equational theory.

Example 2.4.1. Consider a signature with constants S, K, I and with the binary infix symbol '$\cdot$'. The equational theory induced by the following set of equations is undecidable:

$$\Gamma = \{\ I \cdot x \approx x\ \ ,\ \ (K \cdot x) \cdot y \approx x\ \ ,\ \ ((S \cdot x) \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)\ \ \}. \qquad \triangle$$

We will show, however, that this logic is decidable if we just require that the base equational theory is convergent. We may wonder whether the logic would not be decidable by considering decidable equational theories in general, even those for which do not exist any rewriting system decision procedure to decide the validity problem, as is the case of the decidable equational theory induced by the commutative axiom (see [51]). Indeed, it would be very interesting to explore that in the future. For now, it is outside the scope of what we intend to do, as the majority of the equational theories underlying information security examples are generated by convergent rewriting systems. All the more, we take advantage of the rewriting systems underlying equational theories to draw a decidability result. Our decidability proof is actually more informative, as we develop a satisfiability procedure for EqCL by a polynomial reduction to the satisfiability problem for classical propositional logic. That said, our setup is, from now on, that Γ is a convergent equational theory.

2.4.1 Satisfiability

Throughout this subsection we analyze the satisfiability problem for EqCL (SAT-EqCL) and provide a satisfiability algorithm that reduces SAT-EqCL to the SAT problem for classical propositional logic.

The SAT-EqCL problem consists in deciding the existence of a model for a global formula, i.e., given $\delta \in \mathsf{Glob}$, SAT-EqCL decides the existence of an F-structure $(\mathbb{A}, \mathsf{S})$ satisfying δ.

In general, the satisfiability solvers require a particular format for the input formula [65]. The standard input representation for the SAT solvers is the CNF. Following this line, we begin with a detailed analysis of the CNFSAT-EqCL problem for EqCL, the satisfiability problem whose input is restricted to formulas in conjunctive normal form, and then proceed with the satisfiability result.

Moving to the propositional context

To properly describe the algorithm that reduces SAT-EqCL to SAT, we need to translate local formulas to the propositional context. Hence, let us consider a set of propositional symbols corresponding to equations between nominal terms $\mathsf{Eq}(N)^{\mathsf{p}} = \{\mathsf{p}_{t_1 \approx t_2} \mid t_1, t_2 \in T(N)\}$, and then define the translation of a local formula $\varphi \in \mathsf{Loc}$ to a propositional formula prop_φ inductively, by:

- if φ is of the form $t_1 \approx t_2$, prop_φ is precisely $\mathsf{p}_{t_1 \approx t_2}$,

- if φ is of the form $\neg\psi$ then prop_φ is $\neg\mathsf{prop}_\psi$,

- if φ is of the form $\varphi_1 \wedge \varphi_2$ then prop_φ is $\mathsf{prop}_{\varphi_1} \wedge \mathsf{prop}_{\varphi_2}$.

Furthermore, since the satisfiability of EqCL will be reduced to the satisfiability problem of classical propositional logic, we must import the algebraic requirements to the propositional context. For this purpose, assume that we want to test the satisfiability of a global formula $\delta \in \mathsf{Glob}$ and let $\mathsf{RelTerm}^\delta \subseteq T(N)$ represent the set of relevant nominal terms for δ. $\mathsf{RelTerm}^\delta$ should embrace $\mathsf{subterms}(\delta)$, their normal forms with respect to the convergent rewriting system R underlying Γ and still the equational theory instantiated on them:

$$\mathsf{RelTerm}^\delta = \mathsf{subterms}(\delta) \cup \{t{\downarrow} | \, t \in \mathsf{subterms}(\delta)\} \cup \mathsf{subterms}(\Delta^{\approx}),$$

where $\Delta^{\approx} = \{\sigma(t) \approx \sigma(t') \mid (t \to t') \in R, \sigma \in \mathsf{subterms}(\delta)^X\}$.

The propositional symbols of interest are those that represent equations between terms in $\mathsf{RelTerm}^\delta$, and are gathered in the set

$$\mathcal{B}^\delta = \left\{\mathsf{p}_{t_1 \approx t_2} \mid t_1, t_2 \in \mathsf{RelTerm}^\delta\right\}. \tag{2.6}$$

Equational statements must obey some relations, to be imposed on their representatives. These relations are established in Φ^δ and correspond to a reflexivity-like property incorporating the equational theory, symmetry, transitivity and congruence:

$$\begin{aligned}\Phi^\delta = &\{\mathsf{p}_{t \approx t\downarrow} \mid t \in \mathsf{RelTerm}^\delta\} \cup \{\mathsf{p}_{t_1 \approx t_2} \to \mathsf{p}_{t_2 \approx t_1} \mid t_1, t_2 \in \mathsf{RelTerm}^\delta\} \cup \\ &\{\mathsf{p}_{t_1 \approx t_2} \wedge \mathsf{p}_{t_2 \approx t_3} \to \mathsf{p}_{t_1 \approx t_3} \mid t_1, t_2, t_3 \in \mathsf{RelTerm}^\delta\} \cup \\ &\{\mathsf{p}_{t_1 \approx t'_1} \wedge \ldots \wedge \mathsf{p}_{t_n \approx t'_n} \to \mathsf{p}_{f(t_1,\ldots,t_n)\downarrow \approx f(t'_1,\ldots,t'_n)\downarrow} \mid t_1, t'_1, \ldots, t_n, t'_n, f(t_1,\ldots,t_n)\downarrow, f(t'_1,\ldots,t'_n)\downarrow \in \mathsf{RelTerm}^\delta\}.\end{aligned} \tag{2.7}$$

Note that for each $t \in \mathsf{RelTerm}^\delta$, $\mathsf{p}_{t \approx t}$ is a propositional consequence of Φ^δ.

We should emphasize that, since $\mathsf{subterms}(\delta)$ has linear size on the length of δ and the equational theory is convergent, $\mathsf{RelTerm}^\delta$ is well defined and has polynomial size on the length of δ. Denoting $\left|\mathsf{RelTerm}^\delta\right| = k$, Φ^δ has at most $k + k^2 + k^3 + k^{2a+2}$ elements, where a is the maximum arity of the function symbols occurring in $\mathsf{RelTerm}^\delta$. We drop the superscript δ when it is clear from context.

CNFSAT-EqCL problem

The CNFSAT-EqCL problem consists in deciding the existence of a model for a global formula $\delta \in \mathsf{Glob}$ given in conjunctive normal form.

To test the satisfiability of a global formula $\delta \in \mathsf{Glob}$ given in CNF by $\bigwedge\limits_{j=1}^{m}\bigvee\limits_{i=1}^{n_j}\delta_i^j$, one computes the set $\mathcal{B}^\delta$ of propositional symbols described in (2.6), the set Φ^δ of propositional formulas described in (2.7) and then use Algorithm 2.1 to decide whether the given formula is satisfiable or not. Note that each conjunct is a disjunction of literals. Specifying explicitly those components, δ is given by:

$$\bigwedge_{j=1}^{m}\left(\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j\right).$$

The conversion to the propositional context is based on the idea that there should exist as many copies of $\mathcal{B}^\delta$ as the number of existential formulas $\neg\forall\mathsf{Loc}$ occurring in δ. In its very last description, δ counts with $\sum_{j=1}^{m} k_j$ formulas of $\neg\forall\mathsf{Loc}$, so let $\mathcal{B}^*$ carry such number of *labeled copies* of $\mathcal{B}^\delta$:

$$\mathcal{B}^* = \bigcup_{j'=1}^{m}\bigcup_{\ell'=1}^{k_{j'}}\{\mathsf{p}^{[j',\ell']} \mid \mathsf{p} \in \mathcal{B}^\delta\} \cup \mathcal{B}^\delta.$$

When $k_{j'} = 0$, $\bigcup\limits_{\ell'=1}^{k_{j'}}\{\mathsf{p}^{[j',\ell']} \mid \mathsf{p} \in \mathcal{B}^\delta\}$ represents the empty set.

Algorithm 2.1 CNFSAT-EqCL solver based on SAT

```
procedure CNFSATEqCL
    input: global formula $\delta$ given in CNF by $\bigwedge_{j=1}^{m}\left(\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j\right)$
    output: Sat or Unsat depending on whether $\delta$ is satisfiable or not
    propositional symbols: $\bigcup_{j'=1}^{m}\bigcup_{\ell'=1}^{k_{j'}}\{\mathsf{p}^{[j',\ell']} \mid \mathsf{p} \in \mathcal{B}^\delta\} \cup \mathcal{B}^\delta$
    $Q := \bigwedge_{\phi\in\Phi^\delta}\left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}}\phi^{[j',\ell']} \wedge \phi\right)$          ▷ incorporate $\Phi^\delta$ in $Q$
    for $j = 1$ to $m$ do          ▷ define the $j^{th}$ conjunct
        $Q_j := \bot$
        for $s = 1$ to $n_j$ do          ▷ incorporate each $\forall\psi_s^j$ in $Q_j$
            $Q_j := Q_j \vee \left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}}\mathsf{prop}_{\psi_s^j}^{[j',\ell']} \wedge \mathsf{prop}_{\psi_s^j}\right)$
        for $\ell = 1$ to $k_j$ do          ▷ incorporate each $\neg\forall\varphi_\ell^j$ in $Q_j$
            $Q_j := Q_j \vee \neg\mathsf{prop}_{\varphi_\ell^j}^{[j,\ell]}$
    $Q := Q \wedge \bigwedge_{j=1}^{m} Q_j$          ▷ propositional formula $Q$ corresponding to $\delta$
    return sat_solver($Q$)          ▷ return Sat if $Q$ is satisfiable and Unsat otherwise
```

Given a global formula $\delta \in \mathsf{Glob}$ written in conjunctive normal form, the CNFSAT-EqCL solver tests the satisfiability of δ by reduction to a SAT solver, which is represented in Algorithm 2.1 by an auxiliary procedure sat_solver that returns Sat or Unsat, depending on whether the propositional formula given as input is satisfiable or not. Notice that we have fixed a convergent equational theory Γ, so that the sets of propositional formulas $\mathcal{B}^{\delta}$ and Φ^{δ} are well defined and have polynomial size on the length of δ. Recall once again that each conjunct is written as a disjunction of elements from $\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}$. Satisfying an element of the form $\forall\varphi$ imposes that φ must be verified in all possible outcome, whereas satisfying a formula as $\neg\forall\varphi$ requires that at least one possible outcome satisfies $\neg\varphi$. Therefore, our reduction to the propositional context must carry this sensitivity. In this way, the satisfiability of those conjuncts is tested using several labeled copies of propositional symbols (one copy for each literal of the form $\neg\forall\mathsf{Loc}$), as if they had embedded several valuations (exactly one for each literal in $\neg\forall\mathsf{Loc}$). The labels are extended from the propositional symbols to the propositional formulas as expected. When the resulting propositional formula is satisfiable, we conclude that δ is also satisfiable.

Let us illustrate the satisfiability procedure in a small example.

Example 2.4.2. Recall the signature F^{DY} and the equational theory Γ^{DY} introduced in Example 2.1.7. Let us use Algorithm 2.1 to test the satisfiability of the CNF formula:

$$\forall(m \approx k) \wedge \neg\forall(\{\{n\}_k\}_m^{-1} \approx \pi_2(a,n)). \tag{2.8}$$

Note that, in this case, Q is simply a conjunction of several propositional formulas. We reveal just a few of them in order to conclude that the SAT solver would return Unsat in this specific case. From the construction, $Q_1 \wedge Q_2$ would look like

$$\mathsf{prop}_{m\approx k} \wedge \mathsf{prop}^{[2,1]}_{m\approx k} \wedge \neg\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)}.$$

But then, recall the presence of Φ^{δ}, which would imply that $\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_k^{-1}\approx n}$ and $\mathsf{prop}^{[2,1]}_{\pi_2(a,n)\approx n}$ holds. By the transitivity representative in Φ^{δ} we conclude that $\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_k^{-1}\approx\pi_2(a,n)}$ is derivable. The propositional representative of congruence, also allows us to conclude that, since $\mathsf{prop}^{[2,1]}_{m\approx k}$ holds, $\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_m^{-1}\approx\{\{n\}_k\}_k^{-1}}$ must also hold. By transitivity, one concludes that $\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)}$ holds, which contradicts Q_2, $\neg\mathsf{prop}^{[2,1]}_{\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)}$.

Hence, the formula (2.8) is not satisfiable. $\triangle$

Lemma 2.4.3. *If Γ is a convergent equational theory, a global formula $\delta \in \mathsf{Glob}$ in CNF is satisfiable iff Algorithm 2.1 returns* Sat.

Proving this Lemma requires showing that satisfiability at the propositional level carries over to EqCL. For this purpose, given $\delta \in \mathsf{Glob}$, we define a translation of outcomes with values in an F-algebra $\mathbb{A}$ with carrier set A to valuations in the propositional context, and vice-versa. For the first kind of translation, let us denote by $v_{(\cdot)}$ the transformation of outcomes into valuations, $v_{(\cdot)} : A^N \to \{0,1\}^{\mathcal{B}}$ such that, given $\rho \in A^N$, the corresponding valuation $v_\rho : \mathcal{B} \to \{0,1\}$ is defined by:

$$v_\rho(\mathsf{p}_{t_1 \approx t_2}) = 1 \text{ iff } [\![t_1]\!]^\rho_{\mathbb{A}} = [\![t_2]\!]^\rho_{\mathbb{A}}. \tag{2.9}$$

This translation is sound and complete. The following Lemma is easily proved by induction on φ.

Lemma 2.4.4. *For each $\varphi \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and $\rho \in A^N$, $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi$ iff $v_\rho\left(\mathsf{prop}_\varphi\right) = 1$.*

For the second kind of translation, we denote by $[\cdot]$ the transformation of valuations into outcomes $[\cdot] : \{0,1\}^{\mathcal{B}} \to \wp(A^N)$ such that, given $v \in \{0,1\}^{\mathcal{B}}$,

$$[v] \quad = \quad \left\{\rho \in A^N \mid v_\rho \cong v\right\} \tag{2.10}$$

where v_ρ was defined in (2.9) and $\cong$ represents equality of functions. The role of this translation is simply to gather all the outcomes that would lead to the same valuation. To prove that this translation is sound and complete we need the following auxiliary result.

Lemma 2.4.5. *For any terms $t_1, t_2 \in \mathsf{subterms}(\delta)$ and valuation $v \in \{0,1\}^{\mathcal{B}}$ such that $[v] \neq \varnothing$, $v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$ iff $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} t_1 \approx t_2$ for every $\rho \in [v]$.*

Proof. Let $t_1, t_2 \in \mathsf{subterms}(\delta)$ and $v \in \{0,1\}^{\mathcal{B}}$ be a valuation such that $[v] \neq \varnothing$. For the direct implication assume that $v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$ and note that for each $\rho \in [v]$ we have $v_\rho \cong v$, which implies that $v_\rho\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$. By definition of $v_{(\cdot)}$ it is equivalent to $[\![t_1]\!]^\rho_{\mathbb{A}} = [\![t_2]\!]^\rho_{\mathbb{A}}$, and therefore to $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} t_1 \approx t_2$. Reciprocally, assume that for every $\rho \in [v]$ $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} t_1 \approx t_2$, i.e., $v_\rho\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$. This implies that $v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$. □

Lemma 2.4.6. *For any formula $\varphi \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and valuation $v \in \{0,1\}^{\mathcal{B}}$ such that $[v] \neq \varnothing$, $v\left(\mathsf{prop}_\varphi\right) = 1$ iff $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi$ for every $\rho \in [v]$.*

Proof. The proof follows by structural induction on φ and makes use of the previous result.

- if φ is of the form $t_1 \approx t_2$ the result follows from the previous Lemma,
- if φ is of the form $\neg\varphi'$ for some $\varphi' \in \mathsf{Loc}$, then $\varphi' \in \mathsf{subform}(\delta)$ and the following equivalences hold:

$$\begin{array}{ll} & v\left(\mathsf{prop}_{\neg\varphi'}\right) = 1 \\ \text{iff} & v\left(\neg\mathsf{prop}_{\varphi'}\right) = 1 \\ \text{iff} & v\left(\mathsf{prop}_{\varphi'}\right) = 0 \\ \text{iff} & v_\rho\left(\mathsf{prop}_{\varphi'}\right) = 0, \text{ for every } \rho \in [v] \\ \text{iff} & \mathbb{A}, \rho \nVdash_{\mathsf{loc}} \varphi', \text{ for every } \rho \in [v] \\ \text{iff} & \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \neg\varphi', \text{ for every } \rho \in [v] \end{array}$$

- if φ is of the form $\varphi_1 \wedge \varphi_2$ for some $\varphi_1 \wedge \varphi_2 \in \mathsf{Loc}$, then $\varphi_1, \varphi_2 \in \mathsf{subform}(\delta)$ and we have the following equivalences:

$$\begin{array}{ll} & v\left(\mathsf{prop}_{\varphi_1 \wedge \varphi_2}\right) = 1 \\ \text{iff} & v\left(\mathsf{prop}_{\varphi_1} \wedge \mathsf{prop}_{\varphi_2}\right) = 1 \\ \text{iff} & v\left(\mathsf{prop}_{\varphi_1}\right) = 1 \text{ and } v\left(\mathsf{prop}_{\varphi_2}\right) = 1 \\ \text{iff} & v_\rho\left(\mathsf{prop}_{\varphi_1}\right) = 1 \text{ and } v_\rho\left(\mathsf{prop}_{\varphi_2}\right) = 1, \text{ for every } \rho \in [v] \\ \text{iff} & \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_1 \text{ and } \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_2, \text{ for every } \rho \in [v] \\ \text{iff} & \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi_1 \wedge \varphi_2, \text{ for every } \rho \in [v] \ . \end{array}$$ □

The proof of Lemma 2.4.3 needs one more technical Lemma, that will later ensure that there always exist at least one outcome corresponding to each valuation in the propositional side.

Lemma 2.4.7. *Given a valuation $v \in \{0,1\}^{\mathcal{B}}$ that satisfies Φ^δ, we have*

$$[v] = \left\{\rho \in (T(N)_{/\equiv})^N \mid v_\rho \cong v\right\} \neq \varnothing,$$

where $\equiv$ is the congruence relation on $T(N)$ generated by the following rule:

$$\textit{given } s \approx s' \in \Gamma \textit{ and } \sigma \in T(N)^X,\ \sigma(s) \equiv \sigma(s').$$

Proof. Let $v \in \{0,1\}^{\mathcal{B}}$ be a valuation satisfying Φ^δ and let us define a congruence relation $\equiv_v \subseteq (T(N)_{/\equiv} \times T(N)_{/\equiv})$ generated by the rule:

$$\text{for every } t_1, t_2 \in \mathsf{RelTerm}, [t_1]_\equiv \equiv_v [t_2]_\equiv \ \text{ iff } \ v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1.$$

Note that $\equiv_v$ is compatible with $\equiv$, in the sense that $\equiv_v$ is well defined and the definition does not depend on the representative element of each equivalence class of $\equiv$. Indeed, given $t_1, t_2, t_1', t_2' \in \mathsf{RelTerm}$ such that

$$t_1' \in [t_1]_\equiv \tag{2.11}$$

and

$$t_2' \in [t_2]_\equiv \tag{2.12}$$

we easily prove that if $v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$ then $v\left(\mathsf{p}_{t_1' \approx t_2'}\right) = 1$. For this purpose, note that by (2.11) we know that $t_1 \downarrow = t_1' \downarrow$. Since $t_1, t_1' \in \mathsf{RelTerm}$, then $t_1 \downarrow, t_1' \downarrow \in \mathsf{RelTerm}$ as well. Additionally, $\mathsf{p}_{t_1 \approx t_1\downarrow}$ and $\mathsf{p}_{t_1' \approx t_1'\downarrow}$ are propositional consequences of Φ^δ. Since v satisfies Φ^δ, by symmetry and transitivity, we are now able to conclude that $v\left(\mathsf{p}_{t_1 \approx t_1'}\right) = 1$, which together with $v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1$ implies that $v\left(\mathsf{p}_{t_1' \approx t_2}\right) = 1$. A similar reasoning can be done from (2.12) to conclude that $v\left(\mathsf{p}_{t_1' \approx t_2'}\right) = 1$.

Let $[[t]_{\equiv}]^{*}_{\equiv_v}$ be a representative for the equivalence class $[[t]_{\equiv}]_{\equiv_v}$ and consider the outcome

$$\begin{array}{rccl} \rho^v : & N & \to & T(N)_{/\equiv} \\ & n & \mapsto & [[n]_{\equiv}]^{*}_{\equiv_v} \end{array}$$

We now check that $\rho^v \in [v]$, i.e., $v_{\rho^v} \cong v$: given $\mathsf{p}_{t_1 \approx t_2} \in \mathcal{B}$,

$$\begin{array}{rlll} v_{\rho^v}(\mathsf{p}_{t_1\approx t_2}) = 1 & \text{iff} & [\![t_1]\!]^{\rho^v}_{\mathbb{A}} = [\![t_2]\!]^{\rho^v}_{\mathbb{A}} & \text{(by definition of } v_{(\cdot)}\text{)} \\ & \text{iff} & [[t_1]_{\equiv}]^{*}_{\equiv_v} = [[t_2]_{\equiv}]^{*}_{\equiv_v} & \text{(by definition of } \rho^v\text{)} \\ & \text{iff} & [t_1]_{\equiv} \equiv_v [t_2]_{\equiv} & \text{(two equivalence classes are either equal or disjoint)} \\ & \text{iff} & v\,(\mathsf{p}_{t_1\approx t_2}) = 1 & \text{(by definition of } \equiv_v\text{)} \\ & \text{iff} & v(\mathsf{p}_{t_1\approx t_2}) = 1. & \end{array}$$

Since $\rho^v \in [v]$, it follows that $[v] \neq \varnothing$. □

Proof of Lemma 2.4.3. Let $\delta \in \mathsf{Glob}$ be any global formula given in CNF. To prove the direct implication, consider an F-structure $(\mathbb{A}, \mathsf{S})$ satisfying δ: $(\mathbb{A}, \mathsf{S}) \Vdash \bigwedge\limits_{j=1}^{m} \bigvee\limits_{i=1}^{n_j} \delta^j_i$. This means that for each $j \in \{1, \ldots, m\}$, $(\mathbb{A}, \mathsf{S}) \Vdash \bigvee\limits_{i=1}^{n_j} \delta^j_i$. Since each δ^j_i is a literal, we can rewrite it as

$$(\mathbb{A}, \mathsf{S}) \Vdash \neg\forall\varphi^j_1 \vee \ldots \vee \neg\forall\varphi^j_{k_j} \vee \forall\psi^j_1 \vee \ldots \vee \forall\psi^j_{n_j}.$$

Note that for each conjunct, $j \in \{1, \ldots, m\}$, at least one of the disjuncts must be satisfied, i.e.,

$$\text{either there exists } \ell \in \{1, \ldots, k_j\} \text{ such that } \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \neg\varphi^j_\ell \text{ for some } \rho \in \mathsf{S}, \tag{2.13}$$

$$\text{or there exists } s \in \{1, \ldots, n_j\} \text{ such that } \mathbb{A}, \rho \Vdash_{\mathsf{loc}} \psi^j_s \text{ for every } \rho \in \mathsf{S}. \tag{2.14}$$

Given $j \in \{1, \ldots, m\}$ such that $k_j > 0$, consider each $\ell \in \{1, \ldots, k_j\}$ and let us denote by $\rho^{\varphi^j_\ell}$ an outcome in the conditions of (2.13) if one exists, and any other outcome of S in case there is no outcome in conditions (2.13), since any other satisfies all the positive literals, as stated in (2.14). Collect all these valuations in the set $V_j = \left\{ v_{\rho^{\varphi^j_1}}, \ldots, v_{\rho^{\varphi^j_{k_j}}} \right\}$. Then, consider any other outcome $\rho \in \mathsf{S}$ and let $V_j = \{v_\rho\}$ for every $j \in \{1, \ldots, m\}$ with $k_j = 0$. The set of all valuations of interest is the set $\mathcal{V} = \bigcup\limits_{j=1}^{m} V_j \cup \{v_\varphi\} \subseteq \{0,1\}^{\mathcal{B}^\delta}$.

Remark 2.4.8. Observe that, for each $j \in \{1, \ldots, m\}$, all the valuations $v \in V_j$ satisfy the propositional formula $\widetilde{Q_j} \coloneqq \bigvee_{i=1}^{n_j} \mathsf{prop}_{\psi^j_i} \vee \bigvee_{\ell=1}^{k_j} \neg\mathsf{prop}_{\varphi^j_\ell}$ by simply noting that it results from (2.13), (2.14) and from Lemma 2.4.4. Furthermore, all these valuations also satisfy the propositional formula $\bigwedge_{\phi\in\Phi} \phi$, as $(\mathbb{A}, \mathsf{S})$ satisfies each instance of $\mathsf{Eq1}$-$\mathsf{Eq4}$ and $\mathsf{E}(\Gamma)$.

Now let us merge all the valuations of $\mathcal{V}$ together into a valuation $v^* : \mathcal{B}^* \to \{0,1\}$ over the set $\mathcal{B}^*$, defined by:

$$\begin{array}{rcl} v^*(\mathsf{p}^{[j',\ell']}) & = & v_{\rho^{\varphi^{j'}_{\ell'}}}(\mathsf{p}) \\ v^*(\mathsf{p}) & = & v_\rho(\mathsf{p}), \end{array}$$

for each $\mathsf{p} \in \mathcal{B}^\delta$, and observe that v^* satisfies:

$$Q := \bigwedge_{\phi\in\Phi}\left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}} \phi^{[j',\ell']}\wedge\phi\right)\wedge\bigwedge_{j=1}^{m}\left[\bigvee_{s=1}^{n_j}\left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}}\mathsf{prop}^{[j',\ell']}_{\psi^j_s}\wedge\mathsf{prop}_{\psi^j_s}\right)\vee\bigvee_{\ell=1}^{k_j}\neg\mathsf{prop}^{[j,\ell]}_{\varphi^j_\ell}\right].$$

To conclude this, note that:

- For each $\phi \in \Phi$, $j' \in \{1,\dots,m\}$ and $\ell' \in \{1,\dots,k_{j'}\}$, by definition of v^* and using Remark 2.4.8 we have $v^*(\phi^{[j',\ell']}) = v_{\rho^{\varphi^{j'}_{\ell'}}}(\phi) = 1$ and $v^*(\phi) = v_\rho(\phi) = 1$.
- For each $j \in \{1,\dots,m\}$, recall observations (2.13) and (2.14) and notice that: either there exists an $\ell \in \{1,\dots,k_j\}$ such that $v^*\left(\neg\mathsf{prop}^{[j,\ell]}_{\varphi^j_\ell}\right) = v_{\rho^{\varphi^j_\ell}}\left(\neg\mathsf{prop}_{\varphi^j_\ell}\right) = 1$, or there exists an $s \in \{1,\dots,n_j\}$ for which all the outcomes satisfy ψ^j_s, and so, for each $j' \in \{1,\dots,m\}$ and $\ell' \in \{1,\dots,k_{j'}\}$, we use the definition of v^* to conclude that $v^*(\mathsf{prop}^{[j',\ell']}_{\psi^j_s}) = v_{\rho^{\varphi^{j'}_{\ell'}}}(\mathsf{prop}_{\psi^j_s}) = 1$ and $v^*(\mathsf{prop}_{\psi^j_s}) = v_\rho(\mathsf{prop}_{\psi^j_s}) = 1$.

Hence, Algorithm 2.1 returns Sat.

Reciprocally, assume that Algorithm 2.1 returns Sat, i.e., there exists a valuation $v^* : \mathcal{B}^* \to \{0,1\}$ that satisfies Q. In particular, v^* satisfies Q_j for each $j \in \{1,\dots,m\}$:

$$v^*\left(\bigvee_{s=1}^{n_j}\left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}}\mathsf{prop}^{[j',\ell']}_{\psi^j_s}\wedge\mathsf{prop}_{\psi^j_s}\right)\vee\bigvee_{\ell=1}^{k_j}\neg\mathsf{prop}^{[j,\ell]}_{\varphi^j_\ell}\right) = 1\,. \tag{2.15}$$

Consider the valuations $\mathcal{V}^* = \{v\} \cup \bigcup_{j=1}^{m}\{v_{[j,1]},\dots,v_{[j,k_j]}\} \subseteq \{0,1\}^{\mathcal{B}^\delta}$ where, for each $j \in \{1,\dots,m\}$ and $\ell \in \{1,\dots,k_j\}$, $v_{[j,\ell]}$ is the projection of v^* over the $[j,\ell]^{th}$ copy of $\mathcal{B}^\delta$ in $\mathcal{B}^*$ and is defined by

$$v_{[j,\ell]}(\mathsf{p}) = v^*(\mathsf{p}^{[j,\ell]}),$$

and v is the projection of v^* over $\mathcal{B}^\delta$ and is defined by

$$v(\mathsf{p}) = v^*(\mathsf{p}),$$

for each $\mathsf{p} \in \mathcal{B}^\delta$. These are the relevant valuations for the remaining construction.

Now we define a model $(\mathbb{A}, \mathsf{S})$ for δ. Begin by defining the quotient F-algebra $\mathbb{A} = \mathbb{T}(N)_{/\equiv}$, with carrier set $A = T(N)_{/\equiv}$, where $\equiv$ is the congruence relation on $T(N)$ generated by the following rule:

$$\text{given } s \approx s' \in \Gamma \text{ and } \sigma \in T(N)^X,\ \sigma(s) \equiv \sigma(s').$$

From a simple observation we find that, given $s \in T(X)$ and $\sigma \in T(N)^X$, $\sigma(s) \equiv \sigma(s\downarrow)$. Besides the definition of $\mathbb{A}$, we need to define S. Let $\mathsf{S} = \{\rho^*_v \mid v \in \mathcal{V}^*\}$ where, for each $v \in \mathcal{V}^*$, $\rho^*_v \in [v]$ is an outcome chosen from $[v]$, whose existence is ensured by Lemma 2.4.7.

To prove that $(\mathbb{A}, \mathsf{S})$ is actually the F-structure we are looking for, we observe that $\mathbb{A}$ satisfies Γ immediately by definition of $\equiv$ and remark that $\varnothing \neq \mathsf{S} \subseteq A^N$ as an immediate consequence of Lemma 2.4.7. Finally, we check that $(\mathbb{A}, \mathsf{S}) \Vdash \bigwedge_{j=1}^{m} \bigvee_{i=1}^{n_j} \delta_i^j$ or, in other words, that for each $j \in \{1, \ldots, m\}$, $(\mathbb{A}, \mathsf{S}) \Vdash \forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j$, which is immediate from (2.15) by simply observing that for each $j \in \{1, \ldots, m\}$:

- either there exists $\ell \in \{1, \ldots, k_j\}$ such that $v^*\left(\neg\mathsf{prop}_{\varphi_\ell^j}^{[j,\ell]}\right) = 1$, i.e., $v_{[j,\ell]}\left(\neg\mathsf{prop}_{\varphi_\ell^j}\right) = 1$ and, by Lemma 2.4.6, $\mathbb{A}, \rho^*_{v_{[j,\ell]}} \Vdash_{\mathsf{loc}} \neg\varphi_\ell^j$ which implies that $(\mathbb{A}, \mathsf{S}) \Vdash \neg\forall\varphi_\ell^j$;
- or there exists $s \in \{1, \ldots, n_j\}$ for which $v^*\left(\bigwedge_{j'=1}^{m} \bigwedge_{\ell'=1}^{k_{j'}} \mathsf{prop}_{\psi_s^j}^{[j',\ell']} \wedge \mathsf{prop}_{\psi_s^j}\right) = 1$, i.e., for every $j' \in \{1, \ldots, m\}$ and $\ell' \in \{1, \ldots, k_{j'}\}$, we have $v_{[j',\ell']}\left(\mathsf{prop}_{\psi_s^j}\right) = 1$ and $v(\mathsf{prop}_{\psi_s^j}) = 1$, which implies that $(\mathbb{A}, \mathsf{S}) \Vdash \forall\psi_s^j$. □

Tseitin-like transformation on EqCL

We described an algorithm to decide the satisfiability of global formulas written in CNF. However, rewriting a formula into conjunctive normal form can lead to an explosion on the length of the formula. Therefore, instead of simply expanding a given formula to CNF, one usually decides the satisfiability of a global formula $\delta \in \mathsf{Glob}$ by analyzing the satisfiability of an equisatisfiable CNF formula.

In classical propositional logic, the satisfiability of a given propositional formula is usually decided by testing the satisfiability of an equisatisfiable formula in conjunctive normal form that is obtain through the Tseitin's transformation [107]. Let us follow the steps for classical propositional logic and import the Tseitin's transformation to EqCL.

The Tseitin-like transformation for EqCL allows us to convert any global formula $\delta \in \mathsf{Glob}$ into an equisatisfiable CNF formula, with only a linear cost on the length of δ. In EqCL, the idea is to introduce additional atoms $\forall(n_1^{\delta'} \approx n_2^{\delta'})$ for every non-atomic subformula δ' of δ, ensuring that $\forall(n_1^{\delta'} \approx n_2^{\delta'}) \leftrightarrow \delta'$ and, in the end, also ensuring that the former formula is satisfied by imposing $\forall(n_1^{\delta} \approx n_2^{\delta})$.

In this sense, given a global formula $\delta \in \mathsf{Glob}$, we consider the set of all subformulas of δ that are not atoms, $\mathsf{subform}(\delta) \setminus \forall\mathsf{Loc}$, and fix a pair of new (distinct) names for each of them. To ease notation, we denote by $\forall\mathsf{at}(\delta')$ the atom corresponding to the subformula $\delta' \in (\mathsf{subform}(\delta) \setminus \forall\mathsf{Loc})$. Furthermore, we abuse notation and also denote an atom $\delta' \in (\mathsf{subform}(\delta) \cap \forall\mathsf{Loc})$ by $\forall\mathsf{at}(\delta')$. In short:

$$\forall\mathsf{at}(\delta') = \begin{cases} \delta' & \text{if } \delta' \in \mathsf{subform}(\delta) \cap \forall\mathsf{Loc} \\ \forall(n_1^{\delta'} \approx n_2^{\delta'}) & \text{if } \delta' \in \mathsf{subform}(\delta) \setminus \forall\mathsf{Loc} \end{cases}$$

where the names in $\left(\bigcup_{\delta' \in \mathsf{subform}(\delta) \setminus \forall \mathsf{Loc}} \{n_1^{\delta'}, n_2^{\delta'}\}\right) \subseteq N$ are new and distinct from each other.

For each non-atomic subformula $\delta' \in (\mathsf{subform}(\delta) \setminus \forall\mathsf{Loc})$, we define the additional conjuncts $\mathsf{tc}(\delta')$ representing the equivalence $\forall\mathsf{at}(\delta') \leftrightarrow \delta'$ in CNF according to the structure of δ':

$\mathsf{tc}(\neg\psi) = (\forall\mathsf{at}(\neg\psi) \vee \forall\mathsf{at}(\psi)) \wedge (\neg\forall\mathsf{at}(\neg\psi) \vee \neg\forall\mathsf{at}(\psi))$;

$\mathsf{tc}(\psi_1 \wedge \psi_2) = (\neg\forall\mathsf{at}(\psi_1 \wedge \psi_2) \vee \forall\mathsf{at}(\psi_1)) \wedge (\neg\forall\mathsf{at}(\psi_1 \wedge \psi_2) \vee \forall\mathsf{at}(\psi_2)) \wedge (\forall\mathsf{at}(\psi_1 \wedge \psi_2) \vee \neg\forall\mathsf{at}(\psi_1) \vee \neg\forall\mathsf{at}(\psi_2))$;

$\mathsf{tc}(\psi_1 \vee \psi_2) = (\forall\mathsf{at}(\psi_1 \vee \psi_2) \vee \neg\forall\mathsf{at}(\psi_1)) \wedge (\forall\mathsf{at}(\psi_1 \vee \psi_2) \vee \neg\forall\mathsf{at}(\psi_2)) \wedge (\neg\forall\mathsf{at}(\psi_1 \vee \psi_2) \vee \forall\mathsf{at}(\psi_1) \vee \forall\mathsf{at}(\psi_2))$.

We define the Tseitin-like transformation on EqCL simply as:

$$\mathsf{tt}(\delta) = \forall\mathsf{at}(\delta) \wedge \bigwedge_{\delta' \in (\mathsf{subform}(\delta) \setminus \forall\mathsf{Loc})} \mathsf{tc}(\delta').$$

Notice that the obtained CNF formula has linear size on the length of δ, since $\mathsf{subform}(\delta)$ has linear size on the length of δ and the transformation $\mathsf{tc}(\cdot)$ increments the length of the formula only by a constant. As corollary of the previous construction we have the following Lemma.

Lemma 2.4.9. *Given $\delta \in \mathsf{Glob}$, there exists an equisatisfiable formula $\delta' \in \mathsf{Glob}$ in conjunctive normal form whose length is linear on the length of δ and can be computed in polynomial time.*

Proof. As we already observed, the Tseitin-like transformation described above converts any $\delta \in \mathsf{Glob}$ into a global formula $\mathsf{tt}(\delta)$ in conjunctive normal form with linear size on the length of δ. This Tseitin-like transformation can be computed in polynomial time as described above.

To check that $\mathsf{tt}(\delta)$ is equisatisfiable to δ, note that each model for δ can be extended to the new names in order to be a model for $\mathsf{tt}(\delta)$ and any model of $\mathsf{tt}(\delta)$ can lead to a model for δ simply by ignoring the values assigned to the additional names introduced by the Tseitin-like transformation. □

Please note that this construction is equivalent to substituting each atom occurring in δ by propositional symbols, applying Tseitin's transformation (in the propositional context) to the resulting propositional formula and afterwards replacing all the new propositional symbols by additional atoms $\forall(n_1 \approx n_2)$, composed of new and independent names n_1, n_2.

Example 2.4.10. Recall Example 2.1.7. Using the Tseitin-like transformation for EqCL, we can obtain an equisatisfiable formula in CNF for

$$\left(\forall(m \approx k) \vee \forall(\{\{n\}_k\}_m^{-1} \approx n)\right) \to \forall\left(\{\{n\}_k\}_m^{-1} \approx \pi_2(a, n)\right)$$

as follows: begin by rewriting the formula without the connective →, introduced by abbreviation, and then identify its subformulas that are not atoms

$$\underbrace{\underbrace{\neg\overbrace{\big(\forall(m\approx k)\vee\forall(\{\{n\}_k\}_m^{-1}\approx n)\big)}^{\delta_1}}_{\delta_2}\ \vee\ \forall\big(\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)\big)}_{\delta}.$$

The CNF formula equisatisfiable to δ is:

$$\mathsf{tt}(\delta)=\forall\mathsf{at}(\delta)\wedge\mathsf{tc}(\delta_1)\wedge\mathsf{tc}(\delta_2)\wedge\mathsf{tc}(\delta),$$

where

$\mathsf{tc}(\delta_1)=(\forall\mathsf{at}(\delta_1)\vee\neg\forall(m\approx k))\wedge(\forall\mathsf{at}(\delta_1)\vee\neg\forall(\{\{n\}_k\}_m^{-1}\approx n)\wedge(\neg\forall\mathsf{at}(\delta_1)\vee\forall(m\approx k)\vee\forall(\{\{n\}_k\}_m^{-1}\approx n)),$
$\mathsf{tc}(\delta_2)=(\forall\mathsf{at}(\delta_2)\vee\forall\mathsf{at}(\delta_1))\wedge(\neg\forall\mathsf{at}(\delta_2)\vee\neg\forall\mathsf{at}(\delta_1)),$
$\mathsf{tc}(\delta)=(\forall\mathsf{at}(\delta)\vee\neg\forall\mathsf{at}(\delta_2))\ \wedge\ (\forall\mathsf{at}(\delta)\vee\neg\forall(\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)))\wedge$
$\wedge,(\neg\forall\mathsf{at}(\delta)\vee\forall\mathsf{at}(\delta_2)\vee\forall(\{\{n\}_k\}_m^{-1}\approx\pi_2(a,n)))\,.$ △

SAT-EqCL **problem**

In the more generic context, we are looking for a procedure to decide the SAT-EqCL problem, i.e., to decide the existence of a model for a given global formula $\delta\in\mathsf{Glob}$. As already noted, the classical expansion of δ to CNF can lead to an exponential explosion on the length of the formula. For this reason, in general, the more efficient way to reduce the SAT-EqCL problem to CNFSAT-EqCL is not by looking for a global formula in CNF equivalent to δ, but by exploring the Tseitin-like transformation for EqCL and come up with an equisatisfiable formula.

Theorem 2.4.11. *If* Γ *is a convergent equational theory then the* SAT-EqCL *problem is decidable.*

Proof. Given a global formula $\delta\in\mathsf{Glob}$, we use the Tseitin-like transformation for EqCL to convert δ into an equisatisfiable formula $\mathsf{tt}(\delta)$ in conjunctive normal form. Then, we run the CNFSAT-EqCL solver presented in Algorithm 2.1 on $\mathsf{tt}(\delta)$. If CNFSAT-EqCL returns Sat then $\mathsf{tt}(\delta)$ has a model, and so δ has a model; if it returns Unsat, then $\mathsf{tt}(\delta)$ is unsatisfiable and δ is also unsatisfiable. □

Note that, alternatively, we could have chosen, as usual for the satisfiability solvers, to reduce the SAT-EqCL problem to a 3 CNFSAT-EqCL solver that, instead of accepting any CNF formula as input, would require the formula to have at most 3 disjuncts on each conjunct. In fact, also the Tseitin-like transformation for EqCL transforms any global formula into a CNF formula with this format.

2.4.2 Validity

Decidability of EqCL follows as an immediate corollary of the satisfiability result.

Theorem 2.4.12. *If* Γ *is a convergent equational theory then* EqCL *is decidable.*

Proof. Since the deduction meta-theorem holds in EqCL, given a finite set $\Delta \subseteq \mathsf{Glob}$ and a formula $\varphi \in \mathsf{Glob}$, proving that $\Delta \vdash_\Gamma \varphi$ is equivalent to proving that $\vdash_\Gamma ((\bigwedge_{\psi\in\Delta} \psi) \to \varphi)$, so we proceed by checking the decidability of the validity problem.

Let $\delta \in \mathsf{Glob}$ be an arbitrary formula, and let us decide whether $\vdash_\Gamma \delta$ or $\nvdash_\Gamma \delta$ by testing the satisfiability of $\neg\delta$: if $\neg\delta$ is satisfiable, since EqCL is sound, we conclude that $\nvdash_\Gamma \delta$; if $\neg\delta$ is not satisfiable, completeness implies that $\vdash_\Gamma \delta$. □

Let us recall Example 2.4.2 to illustrate our decision procedure.

Example 2.4.13. Consider, once again, the signature F^{DY} and the equational theory Γ^{DY}. Our decision procedure allows us to conclude that the global formula

$$\forall(m \approx k) \to \forall(\{\{n\}_k\}_m^{-1} \approx \pi_2(a, n)).$$

is a theorem of EqCL, provided that in Example 2.4.2 we proved that its negation is unsatisfiable. △

2.4.3 Complexity

The satisfiability result highlights a way of reducing the SAT-EqCL problem to the SAT problem for classical propositional logic, under the assumption that Γ is a convergent equational theory. Actually, our analysis revealed a reduction from SAT-EqCL to CNFSAT-EqCL and, furthermore, from CNFSAT-EqCL to SAT. Given the satisfiability result, we explored soundness and completeness of EqCL to derive the decidability result.

Capitalizing on the polynomial complexity of these reductions and on the complexity of SAT, we analyze complexity of CNFSAT-EqCL and SAT-EqCL. In this context, let us denote by P the class of problems solved in polynomial time by a Turing machine, by NP the class of problems solved in non-deterministic polynomial time by a Turing machine and by coNP the class of problems whose complement is in NP [12, 90].

The complexity of CNFSAT-EqCL

The CNFSAT-EqCL solver presented in Algorithm 2.1 exhibits a way to transform a global formula δ given in CNF by $\bigwedge_{j=1}^{m}\left(\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j\right)$ into a propositional formula with length

$$\left(|\Phi^\delta| \cdot \left(\sum_{j=1}^{m} k_j + 1\right) + \sum_{j=1}^{m}\left(n_j \cdot \left(\sum_{j=1}^{m} k_j + 1\right) + k_j\right)\right) \cdot L\,,$$

where L is the maximum length of the propositional formulas involved in the conjunctions and disjunctions that compose Q through the several steps of Algorithm 2.1 and which is at most the length of δ. Since Φ^δ has polynomial size on the length of δ, provided that Γ is given by means of a convergent rewriting system, the obtained propositional formula has polynomial size on the length of δ. Notice that the length of the formula does not exceed $(m \cdot E \cdot k^{max\{2a+2,3\}} + G \cdot E \cdot m^2) \cdot |\delta|$, where m is the number of conjuncts of the CNF formula δ, G and E are the maximum number of positive and negative literals among all the conjuncts, $k = |\mathsf{RelTerm}^\delta|$ and a is the maximum arity of the function symbols occurring in $\mathsf{RelTerm}^\delta$. Algorithm 2.1 exhibits a polynomial reduction from CNFSAT-EqCL to SAT.

The complexity result for the satisfiability problem CNFSAT-EqCL is parametric and also depends on the complexity of determining normal forms for terms with respect to the equational specification of the algebraic basis, which are fundamental to obtain the set of relevant terms $\mathsf{RelTerm}^\delta$. The complexity of CNFSAT-EqCL is the same as for SAT as long as the complexity of computing normal forms with respect to Γ (call it the $\Gamma{\downarrow}$-*problem*) is in P.

Corollary 2.4.14. *Assuming that Γ is a convergent equational theory whose $\Gamma{\downarrow}$-problem is in* P*, then the satisfiability problem* CNFSAT-EqCL *is in* NP *and the validity problem for CNF formulas in* EqCL *is in* coNP*.*

Note that when the rewriting system underlying the equational theory Γ is subterm convergent, the complexity class of the $\Gamma\downarrow$-problem is in P. In fact, every term rewrites to a strict subterm in each rewriting step, so that, in the worst case, a term t takes $|\mathsf{subterms}(t)|$ steps until reaching its normal form, which is linear on the length of t. Since the unification algorithm also takes linear time (see [76, 92]), it follows that in this case the $\Gamma\downarrow$-problem is actually in P.

We should stress that SAT can obviously be modeled in EqCL. For this purpose, one should simply assign an atom $\forall(t_1 \approx t_2)$ composed by two fresh terms t_1, t_2 to each propositional symbol occurring in the SAT problem.

Corollary 2.4.15. *If Γ is a subterm theory, then* CNFSAT-EqCL *is* NP*-complete.*

We conclude that under the assumption that Γ is a convergent equational theory whose $\Gamma\downarrow$-problem is in P, the satisfiability problem CNFSAT-EqCL has the same complexity as CNFSAT (the satisfiability problem for classical propositional logic whose input is given in conjunctive normal form). Obviously, it would also be the case for 3 CNFSAT-EqCL:

3 CNFSAT-EqCL lies on the same complexity class as 3CNFSAT (the satisfiability problem for classical propositional logic whose input is given as a conjunction of disjunctions of at most 3 propositional symbols or their negations).

Recall that, in contrast to 3CNFSAT, 2CNFSAT is known to be in P [101]. However, the CNFSAT-EqCL solver presented in Algorithm 2.1 does not allow us to conclude that 2 CNFSAT-EqCL is also in P. This limitation relies on the general format of the innermost local formulas in δ and, consequently, on the corresponding propositional formulas composing each disjunct of Q_j which, in general, do not correspond to propositional symbols nor their negations, so we must not expect the final propositional formula Q to be in 2CNF format.

The complexity of SAT-EqCL

The complexity result for SAT-EqCL follows immediately from the analysis of complexity of the CNFSAT-EqCL problem and from Lemma 2.4.9.

Corollary 2.4.16. *Assuming that Γ is a convergent equational theory whose $\Gamma{\downarrow}$-problem is in* P*, then the satisfiability problem* SAT-EqCL *is in* NP *and the validity problem for* EqCL *is in* coNP.

Proof. The satisfiability procedure presented for EqCL reduces the analysis of the satisfiability of δ to the analysis of the satisfiability of its equisatisfiable formula $\mathsf{tt}(\delta)$ written in CNF. By Lemma 2.4.9, $\mathsf{tt}(\delta)$ has a linear size on the length of δ and can be obtained in polynomial time. Consequently, we found out a linear reduction from SAT-EqCL to CNFSAT-EqCL. By Corollary 2.4.14, we conclude that the SAT-EqCL problem is in NP. Since CNFSAT-EqCL has a polynomial reduction to the SAT problem, we have also obtained a polynomial reduction from SAT-EqCL to SAT. □

Corollary 2.4.17. *If Γ is a subterm theory, then* SAT-EqCL *is* NP*-complete.*

2.5 Applications to Information Security

Now, we illustrate how to analyze offline guessing attacks to cryptographic protocols using EqCL.

2.5.1 Offline Guessing Attacks

To analyze *offline guessing* [18], one assumes that an attacker has observed messages named $m_1, \ldots, m_k$ (terms in some algebra). Typically, the attacker may know exactly that the messages were built as $t_1, \ldots, t_k \in T(N)$, but he just cannot know the concrete values of the random and secret names used to build them. Still, he can try to mount an attack by guessing some weak secret used by the parties executing the protocol. The attack is successful if the attacker can distinguish whether his guess is correct or not.

In the context of EqCL, we can actually be more ambitious and assume that the attacker has the ability to guess several weak secrets $s_1, \ldots, s_n \in N$ and exploit the algebraic properties of the protocol and cryptographic primitives to conclude about the exactness of his guesses $s_1^*, \ldots, s_n^*$. The following definition arises very naturally.

Definition 2.5.1. Let $m_1, \ldots, m_k \in T(N)$ represent the messages exchanged by the parties executing a given cryptographic protocol, and Γ denote the equational specification of the underlying algebraic basis. We are in the presence of an *offline guessing attack* to the protocol if there exists a *recipe* $\varphi \in \mathsf{Loc}$, with $\mathsf{subterms}(\varphi) \subseteq T(\{m_1, \ldots, m_k, s_1^*, \ldots, s_n^*\})$ such that:

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \nvdash_\Gamma \quad \forall \varphi$$

and

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \vdash_\Gamma \quad \forall (s_1^* \approx s_1 \wedge \cdots \wedge s_n^* \approx s_n \to \varphi)\,.$$

The recipe is a formula constructed exclusively from messages observed by the attacker and from guesses for the secret values. Such formula is not derivable in general, but is valid under the assumption that the attacker correctly guessed the secrets, proving to be a reliable recipe for the attacker to check whether he actually guessed the secrets.

Note that this formalization generalizes the original notion of offline guessing [18] as it considers multiple guesses whilst allowing the recipe to be more complex than a simple equation. Similarly to the original formulation, the collision of cryptographic primitives also leads to offline guessing attacks. For this purpose, note that, given a 2-ary function symbol $\{\cdot\}.$ representing, for instance, the encryption of a message with a symmetric key, can lead to a collision if two different messages m_1, m_2 encrypted with two different keys k_1, k_2 originate the same ciphertext: $\{m_1\}_{k_1} \approx \{m_2\}_{k_2}$. In cases of collisions with the correct values for the secrets, two wrong guesses may compensate each other and, despite the attacker has no way to confirm the correct value of the secrets, constitute an offline guessing attack. We could avoid these cases by taking recipes that were true under the assumption that the guesses were correct, but were false whenever the attacker failed to guess some of the secrets. In EqCL, this more restrictive notion of offline guessing would be formalized by requiring that the attacker would found a recipe φ such that:

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \vdash_\Gamma \quad \forall (s_1^* \not\approx s_1 \wedge \cdots \wedge s_n^* \not\approx s_n \to \neg\varphi)\,,$$

and

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \vdash_\Gamma \quad \forall (s_1^* \approx s_1 \wedge \cdots \wedge s_n^* \approx s_n \to \varphi)\,.$$

However, collisions are usually exploited in very interesting ways to carry out attacks. Under this evidence, we keep the former approach (Definition 2.5.1) to generalize the original notion of offline guessing attacks.

Of course, the task of analysing the existence of an offline guessing attack to a protocol is undecidable in general, as the recipe may be arbitrarily complex. Still, for the Dolev-Yao theory of Example 2.1.7, the equational theory $\mathsf{Th}(\Gamma^{\mathsf{DY}})$ is generated by the convergent rewriting system obtained by orienting the given equations from left to right. Note that the resulting system is subterm convergent, as each rule rewrites a term to a strict subterm. Under such particular conditions, it is known that the problem is decidable, as only a finite number of 'dangerous' recipes need to be tested [2, 3, 18].

Example 2.5.2. Consider the following protocol adapted from Corin and Etalle [43], where $a, b, n_a, p_{ab} \in N$.

$$1.\ a \to b : (a, n_a)$$

$$2.\ b \to a : \{n_a\}_{p_{ab}}$$

In the first step, some party named a sends a message to another party named b in order to initiate some communication session. The message is a pair containing a's name and a random value (*nonce*) named n_a, that a generated freshly, and which is intended to distinguish this request from other, similar, past or future, requests. Upon reception of the first message, b responds by ciphering n_a with a secret password p_{ab} shared with a. When receiving the second message, a can decrypt it and recognize b's response to his request to initiate a session.

It is relatively simple, in this case, to see that the secret shared password p_{ab} is vulnerable to an offline guessing attack. Suppose that the attacker observes the execution of the protocol by parties a and b, and got hold of the two exchanged messages m_1 and m_2. He can now manipulate these messages, using his guess p^*_{ab} of p_{ab}, and come up with recipe $\{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)$. Indeed, only under the correct guess, should the decryption of m_2 with p^*_{ab} coincide with the second projection of m_1, that is, n_a. We can use EQCL to check that, indeed,

$$\forall(m_1 \approx (a, n_a) \land m_2 \approx \{n_a\}_{p_{ab}}) \nvdash_{\Gamma^{\mathsf{DY}}} \forall(\{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

and

$$\forall(m_1 \approx (a, n_a) \land m_2 \approx \{n_a\}_{p_{ab}}) \vdash_{\Gamma^{\mathsf{DY}}} \forall(p^*_{ab} \approx p_{ab} \to \{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

namely using the three $\mathrm{E}(\Gamma^{\mathsf{DY}})$ axioms that encode the equations in Γ^{DY}. △

2.5.2 Privacy on e-voting

Electronic voting protocols constitute a very active research topic nowadays [15, 50, 73, 82, 99]. Obviously, it would be very interesting to obtain a provably secure e-voting protocol, but several concerns arise very naturally, namely with respect to privacy, authentication or anonymity, to name just a few. With respect to privacy, the most fundamental requirement that one can ask for consists in imposing an environment where the intruder cannot distinguish a user's vote from any other possible vote coming from the same voter.

Let us use EqCL for the analysis of privacy in a very simple voting protocol.

Example 2.5.3. Consider the signature F^{DY} extended with a function symbol $h \in \mathrm{F}_1^{\mathsf{DY}}$ representing the hash of a message, and with constants $A, B, C \in \mathrm{F}_0^{\mathsf{DY}}$ representing the possible votes. Let us further consider the set of Horn clauses Γ^{DY} presented in Example 2.1.7.

Then, consider the very simple voting protocol, inspired in [82], where voter a submits his vote v_a by sending its hash as follows:

$$1.\ a \to s : h(v_a)$$

In this case, it is easy to see that the vote v_a is vulnerable to an offline guessing attack. Considering that m_1 stands for the sent message, note that the attacker can come up with a guess v_a^* for v_a and manipulate m_1 to obtain two recipes m_1 and $h(v_a^*)$. Actually, under the correct guess, $h(v_a^*)$ should coincide with m_1. We can use EqCL to conclude that:

$$\forall(m_1 \approx h(v_a)) \nvdash_{\Gamma_h^{\mathsf{DY}}} \forall(m_1 \approx h(v_a^*)),$$

but

$$\forall(m_1 \approx h(v_a)) \vdash_{\Gamma_h^{\mathsf{DY}}} \forall(v_a^* \approx v_a \to m_1 \approx h(v_a^*)).$$

Hence, we conclude that in the presence of this voting protocol the attacker is able to find out users votes by mounting an offline guessing attack and exploring the vulnerability of each vote. △

2.6 Concluding Remarks

In summary, we combined aspects from classical propositional logic, equational logic and quantifiers to came up with a language that allowed us to express equational constraints locally, but also global properties of the set of all intended outcomes. The language of the logic was built on top of a set of names, whose concrete values were not important but were confined to the possible concretizations imposed by the outcomes; up to now, outcomes should be understood as being obtained non-deterministically, however, later on Chapter 4, when probabilities come into play, they will constitute samples of probability distributions, thereby justifying this terminology. Parametrized by a Horn-clause equational specification of the algebraic basis, we have also obtained a sound and complete deductive system for EqCL. The proof of completeness emerged as a harmonious marriage between techniques from equational logic, first-order logic and classical propositional logic. Afterwards, we combined efforts in order to achieve an automated procedure to decide the satisfiability problem for the logic (SAT-EqCL) provided that the algebraic basis was given by means of a convergent rewriting system. For this purpose, we presented a polynomial reduction of CNFSAT-EqCL to SAT and then proved that a Tseitin-like transformation exists for the EqCL logic and could be applied to decide SAT-EqCL. Provided with these fancy reductions, and under the assumption that the algebraic base is a convergent equational theory whose normal forms can be found in polynomial time, we proved that the SAT-EqCL is in NP. It goes without saying that these results could be used to decide the existence of offline guessing attacks whenever the underlying equational theories are subterm convergent, by capitalizing on the results in [2, 3, 18]. We presented examples of offline guessing attacks to cryptographic protocols at the end of the chapter.

Although our decidability results cover a very interesting range of examples, it would be interesting to explore their extension in order to handle decidable equational theories in general, even those for which do not exist any rewriting system decision procedure to decide the validity problem, as is the case of the decidable equational theory induced by the commutative axiom (see [51]).

Chapter 3

Generalized Probabilistic Satisfiability

This chapter has a slightly different taste. Now, we do not envisage to present a logic (this was done with merit of Fagin, Halpern and Megiddo in [58]), but rather to explore a probabilistic satisfiability problem.

For many years, the satisfiability problem for propositional logic (SAT) has been extensively studied both for theoretical purposes (e.g. complexity theory) and for practical purposes. In spite of its NP-completeness [42], modern tools for solving SAT are able to cope with very large problems in a very efficient manner, leading to applications in many different areas and industries [23]. Naturally, people started extending this problem to more expressive frameworks: for instance in Satisfiability Modulo Theories [49], instead of working in propositional logic, one can try to decide if a formula is valid in some specific first-order theory. One other direction is to extend propositional logic with probabilities. The probabilistic satisfiability problem (PSAT) was originally formulated by George Boole [26], and later studied by Nilsson [88]. This problem consists in deciding the satisfiability of a set of assignments of probabilities to propositional formulas. There has been a great effort on the analysis of the probabilistic satisfiability problem and on the development of efficient tools for the automated treatment of this problem [45, 48, 59, 60, 64].

In this chapter we study a Generalized Probabilistic Satisfiability problem (GenPSAT) extending the scope of PSAT by allowing linear combinations of probabilistic assignments of values to propositional formulas. This problem has applications in the analysis of the security of cryptographic protocols and on estimating the probability of success of attacks, as we will see later on Chapter 4. Intuitively, GenPSAT consists in deciding the existence of a probability distribution satisfying a set of classical propositional formulas with probability 1, and a set of linear inequalities involving probabilities of propositional formulas. The GenPSAT problem was previously identified in the context of the satisfiability of the probabilistic logic in [58], where it was also shown to be NP-complete. Here, we explore the computational behaviour of this problem and present a polynomial reduction from GenPSAT to Mixed-Integer Programming, following the lines of [45, 48]. Mixed-Integer Programming (MIP) [91] is a framework to find an optimal solution for a linear objective function subject to a set of linear constraints over real and integer variables. We will exploit the close relation between SAT and MIP [32] in order to reduce GenPSAT problems to suitable MIP problems.

As observed in many NP-complete problems [33], GenPSAT also presents a phase transition behaviour. By solving batches of parametrized random instances, we observe the existence of a threshold splitting a phase where almost every GenPSAT problem is satisfiable, and a phase where almost every GenPSAT problem is not satisfiable. During such transition, the problems become much harder to solve [33].

As the main contribution of this work, we develop the theoretical framework that allows the translation between GenPSAT and MIP problems, which then allows the implementation of a provably correct solver for GenPSAT. This translation is able to encode strict inequalities and disequalities into the MIP context. With the GenPSAT solver in hands, we are able to detect and study the phase transition behaviour.

The chapter is outlined as follows: in Section 3.1 we briefly recall the PSAT problem; in Section 3.2 we carefully define the GenPSAT problem and establish some results on its complexity; Section 3.3 is dedicated to finding a polynomial reduction from GenPSAT to MIP and a prototype tool is provided for an automated analysis of the problem; in Section 3.4 we analyze the presence of phase transition. This work was presented in the Workshop on Logical and Semantic Frameworks with Applications, LSFA 2016, and was submitted for publication, see [29].

3.1 Preliminaries

Let us begin by fixing a set of propositional symbols $\mathcal{P} = \{x_1, \ldots, x_n\}$. Recall from classical propositional logic that a *literal* is either a propositional symbol or its negation, a *clause* is a non-empty disjunction of one or more literals, and a classical propositional formula is any Boolean combination of propositional symbols. A valuation is a map $v : \mathcal{P} \to \{0, 1\}$, which can be extended to the set of classical propositional formulas. Recall that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$.

Throughout this section, let $\mathcal{V}^* = \{v_1, \ldots, v_{2^n}\}$ denote the set of all valuations defined over the propositional symbols of $\mathcal{P}$ with some fixed enumeration. For simplicity, we denote a probability distribution π over $\mathcal{V}^*$ as a probability vector of size 2^n.

Let us denote by *simple probabilistic formula* an expression of the form $\mathsf{Pr}(c) \bowtie q$, where c is a clause, $q \in \mathbb{Q}$, $0 \leq q \leq 1$ and $\bowtie \in \{=, \leq, \geq\}$. Notice that a probability distribution π satisfies a formula $\mathsf{Pr}(c) \bowtie q$ if

$$\sum_{i=1}^{2^n} (v_i(c) \cdot \pi_i) \bowtie q \ \ .$$

A probability distribution π satisfies a set of simple probabilistic formulas if it satisfies each one of them.

We now recall the PSAT problem [59, 64, 88].

Definition 3.1.1 (PSAT problem)**.** Given a set of propositional symbols $\mathcal{P}$ and a set of simple probabilistic formulas $\Sigma = \{\mathsf{Pr}(c_i) \bowtie q_i \mid 1 \leq i \leq k\}$, the Probabilistic Satisfiability problem (PSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies Σ.

The PSAT problem for $\{\mathsf{Pr}(c_i) \bowtie_i q_i \mid 1 \leq i \leq k\}$ can be formulated algebraically as the problem of finding a solution π for the system of inequalities

$$\begin{cases} V\pi \bowtie q \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} \quad ,$$

where V is the $k \times 2^n$ matrix such that $V_{ij} = v_j(c_i)$, i.e., $V_{ij} = 1$ iff the j-th valuation satisfies the i-th clause, $q = [q_i]$ is the k vector of all q_i and $\bowtie = [\bowtie_i]$ is the k vector of all $\bowtie_i$.

The SAT problem can be modeled as a PSAT instance where the entries q_i of the probability vector are all identical to 1. The PSAT problem was shown to be NP-complete [58, 64], even when the clauses consist of the disjunction of only two literals, 2-PSAT.

3.2 GenPSAT problem

We now extend the notion of simple probabilistic formula to handle linear inequalities involving probabilities of propositional formulas. A *probabilistic formula* is an expression of the form

$$\sum_{i=1}^{\ell} (q_i \cdot \mathsf{Pr}(c_i)) \bowtie q \ \ ,$$

where c_i are propositional clauses, $\bowtie \in \{\geq, <, \neq\}$, $\ell \in \mathbb{N}$ and $q_1, \ldots, q_\ell, q \in \mathbb{Q}$. Observe that formulas with the relational symbols $\leq, >, =$ can be obtained by abbreviation. In the case where $\ell = 1$ and $q_1 = 1$, we obtain a simple probabilistic formula. An *atomic probabilistic formula* is a probabilistic formula where each c_i is a propositional symbol, i.e., $c_i \in \mathcal{P}$ for each i.

We say that a probability distribution π *satisfies* a formula $\sum_{i=1}^{\ell}(q_i \cdot \mathsf{Pr}(c_i)) \bowtie q$ if

$$\sum_{i=1}^{\ell}\left(q_i\left(\sum_{j=1}^{2^n} v_j(c_i)\cdot\pi_j\right)\right) \bowtie q \ .$$

A probability distribution π satisfies a set of probabilistic formulas if it satisfies each one of them.

An *instance* of GenPSAT is a pair (Υ, Σ) where Υ is a set of propositional clauses (also called hard constraints) and Σ is a set of probabilistic formulas (soft constraints). We say that a probability distribution π *satisfies* a GenPSAT instance (Υ, Σ) if it satisfies the set of probabilistic formulas

$$\Xi_{(\Upsilon,\Sigma)} = \Sigma \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Upsilon\} \ . \tag{3.1}$$

Definition 3.2.1. Given a GenPSAT instance (Υ, Σ), the *Generalized Probabilistic Satisfiability problem* (GenPSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies (Υ, Σ).

GenPSAT poses a convenient framework for specifying constraints involving different probabilistic formulas. For instance, one may want to impose that $2 \cdot \mathsf{Pr}(A) \leq \mathsf{Pr}(B)$ for two propositional clauses A, B. Such requirements may be very useful in specifying properties of interesting systems but they cannot be easily expressed in the PSAT framework. We now showcase GenPSAT's expressiveness by encoding the Monty Hall problem [100].

Example 3.2.2. The Monty Hall problem is a puzzle where we are faced with the choice of picking one of three doors, knowing that a prize is behind one of them. After our initial choice, the game host opens one of the remaining doors provided that the prize is not behind it, and gives us the choice of switching or keeping the initial guess. The question is: which option is more advantageous?

To model this problem as a GenPSAT instance, let us define the following propositional symbols: P_i holds if the prize is behind door i, X_i holds if our initial choice is door i, H_i holds if the host reveals door i after our initial choice, for $i \in \{1, 2, 3\}$. Since there are only one door with a prize, one initial choice, and one door revealed by the host, we impose the following restrictions:

$$\Upsilon = \left\{ \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (P_i \wedge \neg P_j \wedge \neg P_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (X_i \wedge \neg X_j \wedge \neg X_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (H_i \wedge \neg H_j \wedge \neg H_k) \right\} \ .$$

Furthermore, the host cannot open neither the chosen door nor the door with the prize and so we include the followings constraints in Υ:

$$P_i \to \neg H_i \text{ and } X_i \to \neg H_i \quad \text{ for each } i \in \{1, 2, 3\} \ .$$

We further assume that the prize has uniform probability of being behind each door and that the initial choice is independent of where the prize is:

$$\Sigma = \bigcup_{i,j\in\{1,2,3\}} \left\{ \mathsf{Pr}(P_i) = \frac{1}{3}, \quad \mathsf{Pr}(P_i \wedge X_j) = \frac{1}{3}\,\mathsf{Pr}(X_j) \right\}$$

Concerning the question of which is more advantageous, switching or keeping our initial choice, we encode *winning by switching* as

$$\mathsf{WbS}: \qquad \bigwedge_{i=1}^{3} (P_i \leftrightarrow (\neg X_i \wedge \neg H_i)) \ \ ,$$

and *winning by keeping* as

$$\mathsf{WbK}: \qquad \bigwedge_{i=1}^{3} (P_i \leftrightarrow X_i) \ \ .$$

We want to the decide whether it is always the case that $\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})$, which can be checked by testing the satisfiability of the GenPSAT instance

$$(\Upsilon, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) < \mathsf{Pr}(\mathsf{WbK})\}) \ \ .$$

As expected, this instance is not satisfiable and the instance $(\Upsilon, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})\})$ is satisfiable, allowing us to conclude that it is always advantageous to switch our initial option.

We can take this analysis one step further, and show that the probability of winning by switching is $\frac{2}{3}$ by checking that the instance $(\Upsilon, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \neq \frac{2}{3}\})$ is unsatisfiable and that the instance $(\Upsilon, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) = \frac{2}{3}\})$ is satisfiable. All these instances were checked using the tool we implemented, [30]. △

Notice that the PSAT problem for Σ can be modeled in GenPSAT by considering the instance $(\varnothing, \Sigma)$.

Given a GenPSAT instance (Υ, Σ), where Υ contains m clauses and Σ is composed of k probabilistic formulas, we follow the lines of Nilsson [88] for a linear algebraic formulation and consider a $(k+m)\times 2^n$ matrix $V = [V_{ij}]$, where for each $i \in \{1,\ldots,k+m\}$ and $j \in \{1,\ldots,2^n\}$ V_{ij} is defined from the j^{th} valuation v_j and from the i^{th} probabilistic formula $\sum_{u=1}^{\ell} q_u^i \cdot \mathsf{Pr}(c_u^i) \bowtie_i p_i$ of $\Xi_{(\Upsilon,\Sigma)}$ as follows:

$$V_{ij} = \sum_{u=1}^{\ell} q_u^i \cdot v_j(c_u^i) \ \ .$$

Furthermore, define two vectors of size $k+m$, $q = [q_i]$ and $\bowtie = [\bowtie_i]$. GenPSAT is equivalent to the problem of deciding the existence of a solution π to the system

$$\begin{cases} V\pi \bowtie q \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} \quad . \tag{3.2}$$

Given a set of probabilistic formulas $\Omega = \left\{ \sum_{u=1}^{\ell} q_u^i \cdot v_j(c_u^i) \bowtie_i p_i \mid 1 \leq i \leq k \right\}$ and a set of valuations $\mathcal{V} = \{v_1, \ldots, v_{k'}\}$, we define the $[\Omega, \mathcal{V}]$-*associated matrix* as the $(k+1) \times k'$ matrix $M_{[\Omega,\mathcal{V}]} = [M_{ij}]$ such that

$$M_{k+1,j} = 1 \quad \text{for each } 1 \leq j \leq k'$$

and

$$M_{ij} = \sum_{u=1}^{\ell} q_u^i \cdot v_j(c_u^i) \quad \text{for } 1 \leq i \leq k, \ \ 1 \leq j \leq k' \ .$$

Then, we can rewrite system (3.2) using the $[\Xi_{(\Upsilon,\Sigma)}, \mathcal{V}^*]$-associated matrix V as

$$\begin{cases} V\pi \bowtie q \\ \pi \geq 0 \end{cases} \tag{3.3}$$

We should stress that the GenPSAT problem is a particular case of the satisfiability problem for the probabilistic logic introduced by Fagin, Halpern and Megiddo in [58] and that we presented in Section 1.2. Therefore, it was already proved to be NP-complete (Theorem 1.2.4). Nevertheless, we present a proof for it and use the well-known result from linear programming presented in Lemma 1.2.3.

Theorem 3.2.3 ([58])**.** GenPSAT *is* NP*-complete.*

Proof. We begin by showing that GenPSAT is in NP by providing a polynomial sized certificate. Notice that Lemma 1.2.3 can be extended to rational coefficients simply by normalizing with the greatest denominator. Applying this result to the system (3.3) we conclude that there is a $(k+m+1) \times (k+m+1)$ matrix W, composed of columns of V, whose system

$$\begin{cases} W\pi \bowtie q \\ \pi \geq 0 \end{cases} \tag{3.4}$$

has a solution iff the original system (3.3) has a solution. Furthermore, the obtained solutions from (3.4) can be mapped to solutions of (3.3) by inserting zeros in the appropriate positions. Since the obtained solution from the latter system has $k+m+1$ elements, it constitutes the NP-certificate for the GenPSAT problem.

Furthermore, given that the PSAT problem can be modeled in GenPSAT, it follows that GenPSAT is NP-complete. □

We say that a GenPSAT instance (Υ, Σ) is in *normal form* if Υ is a set of propositional clauses with 3 literals, i.e., Υ can be seen as a 3CNF formula, and Σ is a set of atomic probabilistic formulas.

Lemma 3.2.4. *Given a* GenPSAT *instance* (Υ, Σ) *there exists an instance* (Υ', Σ') *in normal form such that* (Υ, Σ) *is satisfiable iff* (Υ', Σ') *is satisfiable. Moreover,* (Υ', Σ') *is obtained from* (Υ, Σ) *in polynomial time.*

Proof. Let (Υ, Σ) be the GenPSAT instance to be put in normal form. We obtain Σ' by transforming formulas in Σ into atomic probabilistic formulas. For this purpose, let $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(c_i) \bowtie q$ be a formula in Σ and consider the atomic probabilistic formula obtained by replacing (when needed) each clause c_i by a fresh variable y_i,

$$\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) \bowtie q \ .$$

Furthermore, the y_i variable is added to $\mathcal{P}$ and the formula stating the equivalence between y_i and c_i, $(y_i \leftrightarrow c_i)$, is collected in a set Δ.

We are left with the transformation of the formula

$$\bigwedge_{\gamma \in \Upsilon} \gamma \wedge \bigwedge_{(y \leftrightarrow c) \in \Delta} (y \leftrightarrow c)$$

into 3-CNF using Tseitin's transformation [107], which can increase linearly the length of the formula and add new variables to $\mathcal{P}$. The final Υ' is the set of conjuncts of the obtained 3-CNF formula. Since Tseitin's transformation preserves satisfiability of formulas, (Υ, Σ) is satisfiable iff (Υ', Σ') is satisfiable. □

3.3 Reducing GenPSAT to Mixed-Integer Programming

In this section we explore the close relation between satisfaction of propositional formulas and feasibility of a set of linear constraints over binary variables (see [32]). With this, we present a reduction of GenPSAT to Mixed-Integer Programming (MIP), similarly to what was done for PSAT [45] and GPSAT [48]. A MIP problem consists in optimizing a linear objective function subject to a set of linear constraints over real and integer variables. MIP was shown to be NP-complete, see [91]. Observe that this translation to MIP also serves as a proof that GenPSAT is in NP.

3.3.1 Linear Algebraic Formulation for GenPSAT

Lemma 3.3.1. *A* GenPSAT *instance in normal form* (Υ, Σ), *with* $|\Sigma| = k$, *is satisfiable iff there exists a* $(k+1) \times k'$ *matrix* W *of rank* $k' \leq k+1$ *and a set of valuations* $\mathcal{V}_0$ *of size* k' *such that:*

(i) W *is the* $[\Sigma, \mathcal{V}_0]$*-associated matrix*

(ii) $\mathcal{V}_0$ *satisfies* Υ,

(iii) considering $q = [q_1, \dots, q_k, 1]$ *and* $\bowtie = [\bowtie_1, \dots, \bowtie_k, =]$*, the system*

$$\begin{cases} W\pi \bowtie q \\ \pi \geq 0 \end{cases} \tag{3.5}$$

is satisfiable.

Proof. Let (Υ, Σ) be a satisfiable GenPSAT instance in normal form, with $|\Sigma| = k$ and $|\Upsilon| = m$. Then, denoting by V the $[\Xi_{(\Upsilon,\Sigma)}, \mathcal{V}^*]$-associated matrix, the system

$$\begin{cases} V\pi \bowtie q \\ \pi \geq 0 \end{cases}$$

has a solution. And so, using Lemma 1.2.3, there is a $(k + m + 1) \times \ell$ matrix V^*, where $\ell \leq k + m + 1$, and whose system has a positive solution π^*. Notice that the set of valuations underlying V^* certainly satisfies Υ, as $\pi^*_j > 0$ for each $1 \leq j \leq \ell$.

Let W^* be the matrix constructed from V^* by choosing the first k rows (corresponding to the probabilistic formulas in Σ) and the last row (requiring that the solution sums up to one) of V^*. Still, the corresponding system has a positive solution. Using Lemma 1.2.3 once more, we conclude that exists a $(k + 1) \times k'$ matrix W, with $k' \leq k + 1$, whose system has a positive solution $\overline{\pi}$. The solution π for (3.5) is obtained from $\overline{\pi}$ by inserting zeros in the appropriate positions.

Reciprocally, assume that there exists a $(k+1) \times k'$ matrix W of rank $k' \leq k+1$ satisfying (i), (ii), (iii), and let π denote the solution for (3.5). We are looking for a probability distribution π^* satisfying (Υ, Σ). For this purpose, let $\mathcal{V}_0 = \{v_{j_1}, \dots, v_{j_{k'}}\} \subseteq \mathcal{V}$ denote the set of valuations underlying W according to condition ((ii)), and define $\pi^* = [\pi^*_i]$, where

$$\pi^*_i = \begin{cases} \pi_i & \text{if } i \in \{j_1, \dots, j_{k'}\} \\ 0 & \text{otherwise} \end{cases} .$$

The verification that π^* satisfies the GenPSAT instance is now immediate:

- given $\gamma \in \Upsilon$, we check that π^* verifies $\mathsf{Pr}(\gamma) = 1$ by observing that the last equality represented on W on (3.5) leads to $\sum_{s=1}^{k'} \pi_{j_s} = 1$ and so,
$$\sum_{j=1}^{2^n} v_j(\gamma) \cdot \pi^*_j = \sum_{\{j | v_j(\gamma)=1\}} \pi^*_j = \sum_{s=1}^{k'} \pi_{j_s} = 1 \ .$$
- given an atomic probabilistic formula $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) \bowtie q$ in Σ, we recall the definition of π^* and that π is a solution for (3.5) to conclude that
$$\sum_{i=1}^{\ell} q_i \left(\sum_{j=1}^{2^n} v_j(y_i) \cdot \pi^*_j \right) = \sum_{i=1}^{\ell} q_i \left(\sum_{s=1}^{k'} v_{j_s}(y_i) \cdot \pi_{j_s} \right) = \sum_{s=1}^{k'} \left(\sum_{i=1}^{\ell} q_i \cdot v_{j_s}(y_i) \right) \pi_{j_s} \bowtie q \ ,$$
i.e., π^* satisfies the formulas in Σ.

□

3.3.2 Translation to MIP

Regarding Lemma 3.3.1, given a GenPSAT instance (Υ, Σ) in normal form, with $|\Sigma| = k$ and $|\Upsilon| = m$, our goal is now to describe a procedure that encodes the problem of finding a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii), as a MIP problem. We dub this procedure GenToMIP.

Let us denote by $H = [h_{ij}]$ the (still unknown) matrix of size $n \times k'$ whose columns represent the valuations in $\mathcal{V}_0$ evaluated on each propositional symbol of $\mathcal{P}$, i.e., $h_{ij} = v_j(x_i)$ for each $1 \leq i \leq n$ and $1 \leq j \leq k'$. Let $\alpha_1, \dots, \alpha_n$ represent the probability of the propositional symbols $x_1, \dots, x_n$, respectively, and following the reasoning of [45,48] we model the non-linear constraint $\sum_{j=1}^{k'} h_{ij} \cdot \pi_j = \alpha_i$ as a linear inequality

$$\sum_{j=1}^{k'} b_{ij} = \alpha_i \quad , \qquad \text{(val1)}$$

by introducing the extra variables b_{ij} which are subject to the appropriate constraints, namely forcing b_{ij} to be zero whenever $h_{ij} = 0$, and ensuring that $b_{ij} = \pi_j$ whenever $h_{ij} = 1$, i.e.,

$$0 \leq b_{ij} \leq h_{ij} \text{ and } h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j \quad . \qquad \text{(val2)}$$

We ensure that π represents a probability distribution by imposing that

$$\sum_{j=1}^{k'} \pi_j = 1 \quad . \qquad \text{(sums1)}$$

Still, as each valuation of $\mathcal{V}_0$ satisfies Υ, given a clause $\left(\bigvee_{r=1}^{w} x_{i_r} \right) \vee \left(\bigvee_{s=1}^{w'} \neg x_{i'_s} \right)$ of Υ, we generate a linear inequality for each valuation $1 \leq j \leq k'$,

$$\left(\sum_{r=1}^{w} h_{i_r,j} \right) + \left(\sum_{s=1}^{w'} (1 - h_{i'_s j}) \right) \geq 1. \qquad \text{(gamma)}$$

Notice that, if we have a total of m clauses in Υ, we generate $m \times k'$ such inequalities.

In order to verify the satisfiability of probabilistic formulas in the MIP framework, consider an atomic probabilistic formula $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) \bowtie q$ in Σ. Since $\bowtie$ can either be the relational symbol $\geq$, $<$ or $\neq$, we can easily encode the first kind of inequalities as a MIP linear constraint, but should be careful when dealing with the remaining relational symbols.

For atomic probabilistic formulas of the form $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) \geq q$, we generate the linear inequality

$$\sum_{i=1}^{\ell} q_i \cdot \alpha_i \geq q \quad . \qquad (\mathsf{prob}_{\geq})$$

In the case where $\bowtie$ is a strict inequality $<$, we use a specific variable introduced into the MIP problem, say ε, to fix the objective function as the maximization of ε,

$$\mathsf{maximize}\ \varepsilon \tag{obj}$$

and further introduce the linear constraint

$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) + \varepsilon \leq q \ \ . \tag{$\mathsf{prob}_<$}$$

For atomic probabilistic formulas φ of the form $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) \neq q$, i.e.

$$\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(y_i) - q \neq 0, \tag{3.6}$$

we force the left hand side to be either strictly greater or strictly less than zero,

$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) - q < 0 \qquad \text{or} \qquad \sum_{i=1}^{\ell}(q_i \cdot \alpha_i) - q > 0 \ \ .$$

Even though these are linear constraints, the problem would explode if we treated the disjunction. In this sense, notice that, denoting by C a sufficiently large number, say $C = 1 + |q| + \sum_{i=1}^{\ell} |q_i|$, the inequality (3.6) holds if and only if there exists a fresh binary variable z_φ such that the following two strict inequalities hold simultaneously:

$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) - q < C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) + q < C - C \cdot z_\varphi \ \ .$$

Then, we are left with two strict inequalities, thus reducing this analysis to a previous case, from which we obtain the constraints

$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) - q + \varepsilon \ \leq \ C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(q_i \cdot \alpha_i) + q + \varepsilon \ \leq \ C - C \cdot z_\varphi \ \ . \tag{$\mathsf{prob}_{\neq}$}$$

Denoting by $k_\geq$, $k_<$, $k_\neq$ the number of probabilistic formulas in Σ when $\bowtie$ coincides with $\geq$, $<$, $\neq$, respectively, so far we have introduced:

- n constraints ($\mathsf{val1}$),
- $4 \times n \times k'$ constraints ($\mathsf{val2}$),
- 1 constraint ($\mathsf{sums1}$),
- $m \times k'$ constraints (gamma),
- $k_\geq$ constraints ($\mathsf{prob}_\geq$),
- $k_<$ constraints ($\mathsf{prob}_<$),

- $2 \times k_{\neq}$ constraints ($\mathsf{prob}_{\neq}$).

Hence, we have $\mathcal{O}(n + n \times k' + m \times k' + k)$ inequalities over $n \times k'$ binary variables h_{ij}, $n \times k'$ real variables b_{ij}, n real variables $0 \leq \alpha_i \leq 1$, $k_{\neq}$ binary variables z_{φ}, a real variable $\varepsilon \geq 0$ and k' real variables $\pi_j \geq 0$. Because of this, the GenToMIP translation is polynomial.

Proposition 3.3.2. *The* GenToMIP *procedure transforms a* GenPSAT *instance in normal form* (Υ, Σ) *into a* MIP *problem whose size is polynomial on the size of* (Υ, Σ).

We now need to show that the existence of a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii) of Lemma 3.3.1 is equivalent to the feasibility of the MIP problem obtained through GenToMIP with an optimal value $\varepsilon > 0$ (when applicable).

This procedure is presented in Algorithm 3.1, which given a GenPSAT instance, translates it into a MIP problem and then solves the latter appropriately. For that, let us assume that we initialize an empty MIP problem and consider the following auxiliary procedures:

- add_const introduces a linear constraint into the MIP problem,

- set_obj defines the objective function (either as a maximization or as a minimization) when it was previously not defined,

- fresh declares a fresh binary variable into the MIP problem,

- mip_sat returns True or False depending on whether the problem is feasible (and achieves an optimal solution) or not,

- mip_objvalue returns the objective value, when an objective function was set.

Algorithm 3.1 GenPSAT solver based on MIP

```
procedure GenPSAT(props $\{x_i\}_{i=1}^n$, form $\Upsilon$, probform $\Sigma$)
    declare: binary variables: $h_{ij}$, for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
             [0,1]-variables: $\alpha_i$, $\pi_j$ , $b_{ij}$, for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
             real variable: $\varepsilon$
    for $j = 1$ to $k'$ do
        for each $(\bigvee_r x_r) \vee (\bigvee_s \neg x_s)$ in $\Upsilon$ do
            add_const($\sum_r h_{rj} + \sum_s (1 - h_{sj}) \geq 1$)                ▷ (gamma)
    for $i = 1$ to $n$ do
        add_const($\sum_j b_{ij} = \alpha_i$)                                    ▷ (val1)
        for $j = 1$ to $k'$ do
            add_const($0 \leq b_{ij} \leq h_{ij}$)                               ▷ (val2)
            add_const($h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j$)                 ▷ (val2)
    $aux := 0$
    for each $\sum q_i \cdot \mathsf{Pr}(x_i) \bowtie q$ in $\Sigma$ do
        switch($\bowtie$)
            case " $\geq$ " :
                add_const($\sum q_i \cdot \alpha_i \geq q$)                        ▷ (prob$_\geq$)
            case " $<$ " :
                $aux := 1$
                set_obj(max $\varepsilon$)                                         ▷ (obj)
                add_const($\sum q_i \cdot \alpha_i + \varepsilon \leq q$)            ▷ (prob$_<$)
            case " $\neq$ " :
                $aux := 1$
                $z :=$ fresh()                          ▷ $z$ is a fresh binary variable
                $C := 1 + |q| + \sum |q_i|$
                set_obj(max $\varepsilon$)                                         ▷ (obj)
                add_const($\sum q_i \cdot \alpha_i - C \cdot z - \varepsilon \geq q - C$)   ▷ (prob$_\neq$)
                add_const($\sum q_i \cdot \alpha_i - C \cdot z + \varepsilon \leq q$)       ▷ (prob$_\neq$)
    add_const($\sum \pi_i = 1$)                                                  ▷ (sums1)
    if  mip_sat() then
        if ($aux == 0$) or ($aux == 1$ and mip_objvalue() $> 0$) then
            return Sat
    return Unsat
```

Proposition 3.3.3. *A* GenPSAT *instance in normal form* (Υ, Σ) *is satisfiable iff Algorithm 3.1 returns* Sat.

Proof. Let (Υ, Σ) be a satisfiable GenPSAT instance in normal form, and also $\mathcal{V}_0 = \{v_1, \dots, v_{k'}\}$ and $\rho = [\rho_i]$ represent a set of valuations and a probability distribution given by Lemma 3.3.1 which satisfy conditions (i)-(iii). Then, consider the following values and afterwards let us check that they constitute an optimal solution for the MIP problem constructed at Algorithm 3.1: for each $1 \leq i \leq n$ and $1 \leq j \leq k'$, let

$$\begin{aligned} h^*_{ij} &= v_j(x_i), \\ b^*_{ij} &= h^*_{ij} \cdot \rho_j, \\ \pi^*_j &= \rho_j, \\ \alpha^*_i &= \sum_{\{j | v_j(x_i)=1\}} \rho_j, \\ \varepsilon^* &= \mathsf{min}\ \Delta, \end{aligned}$$

where
$$\begin{aligned} \Delta = \ & \{q - \textstyle\sum_{i=1}^{\ell}(q_i \cdot \alpha^*_i) \mid (\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) < q) \in \Sigma\} \ \cup \\ \cup \ & \{C \cdot z^*_\varphi + q - \textstyle\sum_{i=1}^{\ell}(q_i \cdot \alpha^*_i) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \neq q\} \ \cup \\ \cup \ & \{C - C \cdot z^*_\varphi - q + \textstyle\sum_{i=1}^{\ell}(q_i \cdot \alpha^*_i) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \neq q\}, \end{aligned}$$

and, for each atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \neq q$,

$$z^*_\varphi = \begin{cases} 0, & \text{if } \sum_{i=1}^{\ell} q_i \cdot \alpha^*_i < q \\ 1, & \text{if } \sum_{i=1}^{\ell} q_i \cdot \alpha^*_i > q \end{cases} .$$

Now let us check that each linear constraint introduced into the MIP problem at Algorithm 3.1 is satisfied.

(gamma) $\{h^*_{ij}\}$ satisfy the constraints modeling Υ since each $v \in V_0$ satisfies Υ.

(val1) By definition of $\{b^*_{ij}\}$ and $\{h^*_{ij}\}$, we actually have

$$\sum_{j=1}^{k'} b^*_{ij} = \sum_{j=1}^{k'} h^*_{ij} \cdot \rho_j = \sum_{j=1}^{k'} v_j(x_i) \cdot \rho_j = \sum_{\{j | v_j(x_i)=1\}} \rho_j = \alpha^*_i \quad .$$

(val2) Since $0 \leq v_j(x_i) \leq 1$ and $0 \leq \rho_j \leq 1$ we immediately have $0 \leq b^*_{ij} \leq h^*_{ij}$.

For the other inequality, recall that $h^*_{ij} = v_j(x_i)$ and that $\pi^*_j = \rho_j$ and note that:

- if $h^*_{ij} = 0$ then $b^*_{ij} = 0$ and, since $\pi^*_j \leq 1$, it follows that $\pi^*_j - 1 \leq b^*_{ij} \leq \pi^*_j$, i.e.,

$$h^*_{ij} - 1 + \pi^*_j \leq b^*_{ij} \leq \pi^*_j$$

- if $h^*_{ij} = 1$ then $b^*_{ij} = \pi^*_j$ and so $\pi^*_j \leq b^*_{ij} \leq \pi^*_j$, i.e., $h^*_{ij} - 1 + \pi^*_j \leq b^*_{ij} \leq \pi^*_j$

(sums1) Since $\pi_j^* = \rho_j$, we immediately conclude that $\sum\limits_{j=1}^{k'} \pi_j^* = 1$.

To check that the probabilistic formulas are satisfiable, just note that, given a probabilistic formula $(\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \bowtie q) \in \Sigma$,

$$\sum_{i=1}^{\ell} q_i \cdot \alpha_i^* = \sum_{i=1}^{\ell} q_i \left(\sum_{\{j | v_j(x_i)=1\}} \rho_j \right) = \sum_{i=1}^{\ell} q_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right).$$

(prob$_{\geq}$) Let $(\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \geq q) \in \Sigma$ and notice that since ρ satisfies conditions (i), (ii), (iii) it satisfies all the probabilistic formulas in Σ, and so we have $\sum_{i=1}^{\ell} q_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right) \geq q$, which implies that $\sum_{i=1}^{\ell} q_i \cdot \alpha_i^* \geq q$.

(prob$_{<}$) Now, let $(\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) < q) \in \Sigma$ and notice that, in a reasoning very similar to the previous one, we can conclude that $\sum_{i=1}^{\ell} q_i \cdot \alpha_i^* < q$, i.e.

$$q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) > 0. \tag{3.7}$$

But we should also note that, since $\varepsilon^* = \mathsf{min}\ \Delta$, then $\varepsilon^* \leq q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*)$, and so we obtain

$$\sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) + \varepsilon^* \leq \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) + q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) = q\ \ .$$

(prob$_{\neq}$) Finally, let us consider an atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} q_i \cdot \mathsf{Pr}(x_i) \neq q$, and recall once more that since ρ satisfies each probabilistic formula of Σ, we have $\sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) \neq q$, in other words, either $q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) > 0$ or $q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) < 0$. Recall the constant C defined as $C = 1 + |q| + \sum_{i=1}^{\ell} |q_i|$ and the definition of z_φ^* and notice that both

$$C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) > 0 \tag{3.8}$$

and

$$C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) > 0 \tag{3.9}$$

are verified in either of the above cases. Also note that by definition of ε^*, $\varepsilon^* \leq C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*)$ and $\varepsilon^* \leq C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*)$. We now analyze each of the previous cases:

- if $q > \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*)$, then $z_\varphi^* = 0$ and it follows that

  $$\sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) - C \cdot z_\varphi^* - \varepsilon^* \geq \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) - C + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) = q - C,$$

  and further,

  $$\sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) - C \cdot z_\varphi^* + \varepsilon^* \leq \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell} (q_i \cdot \alpha_i^*) = q.$$

- if $q < \sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*)$, then $z_\varphi^* = 1$ and it follows that
$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*) - C \cdot z_\varphi^* - \varepsilon^* \geq \sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*) - C - (C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*)) = q - C,$$
and further,
$$\sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*) - C \cdot z_\varphi^* + \varepsilon^* \leq \sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*) - C + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(q_i \cdot \alpha_i^*) = q.$$

To finish the direct implication, notice that $\varepsilon^* > 0$ as a consequence of (3.7), (3.8) and (3.9), and it takes the maximum possible value since otherwise, let φ_Δ be the formula in Σ which has the minimum value in Δ. Then, if there was a solution with greater objective value it would violate the constraint ($\mathsf{prob}_{\bowtie}$) for φ_Δ.

Reciprocally, assume that Algorithm 3.1 returned Sat, and denote by h_{ij}^*, α_i^*, ε^* and π_j^* the (optimal) solution for the variables h_{ij}, α_i, ε and π_j, for each $1 \leq i \leq n$, $1 \leq j \leq k'$ respectively.

Consider the set of valuations $\mathcal{V}_0 = \{v_1, \dots, v_{k'}\}$ where, for each propositional symbol $x_i \in \mathcal{P}$, $v_j(x_i) = h_{ij}^*$. Due to constraints (gamma) it is immediate to conclude that each valuation satisfies Υ. Then, let the probability distribution π be defined over the set of valuations as the 2^n vector $\pi = [\rho_j]$ where $\rho_j = \pi_j^*$ for $1 \leq j \leq k'$ and $\rho_j = 0$ for $k' < j \leq 2^n$. Note that (sums1) implies that π is a probability vector. The third condition described in Lemma 3.3.1 is deduced by simple inspection of the linear constraints ($\mathsf{prob}_{\geq}$), ($\mathsf{prob}_{<}$), ($\mathsf{prob}_{\neq}$) and (sums1), by definition of the matrix associated to Σ over $\mathcal{V}_0$ and recalling that the optimal value ε^* is such that $\varepsilon^* > 0$. □

As a corollary of the previous propositions, we obtain the following result.

Theorem 3.3.4. *The* GenToMIP *algorithm is a correct translation of* GenPSAT *to a* MIP *problem of polynomial size.*

3.4 Phase Transition

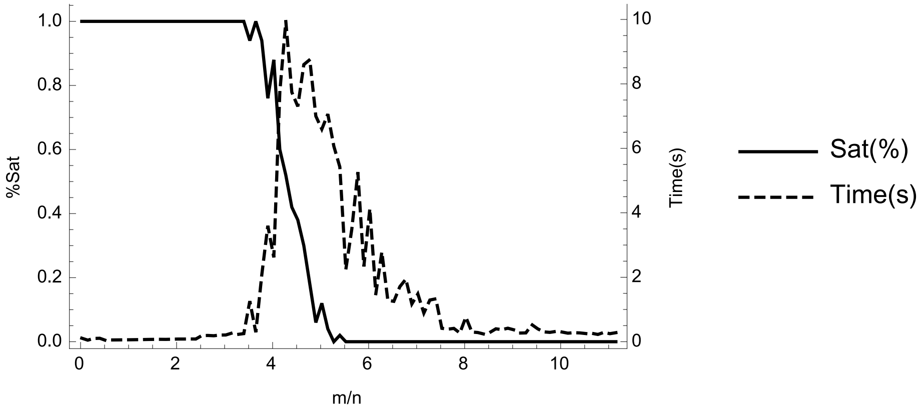
Phase transition is a phenomenon that marks a hardness shift in the solution of instances of a problem. This behaviour was observed in many NP-complete problems [33], among which we highlight 3-SAT [62] and PSAT [59,60].

In this section, we study the GenPSAT phase transition, through an implementation of Algorithm 3.1 and tests comprised of batteries of random instances. For this, we measure the proportion of satisfiable instances as well as the average time the solver spent to solve them. The software was written in Java, and we used Gurobi [68], version 6.5.0, to solve the MIP problem. The machine used for the tests was a Mac Pro at 3,33 GHz 6-Core Intel Xeon with 6 GB of memory. Our implementation is available in [30].

It was noted that, in random 3-SAT instances [62] there is a clear stage where the instances are almost surely satisfiable and one where they are almost surely not satisfiable.

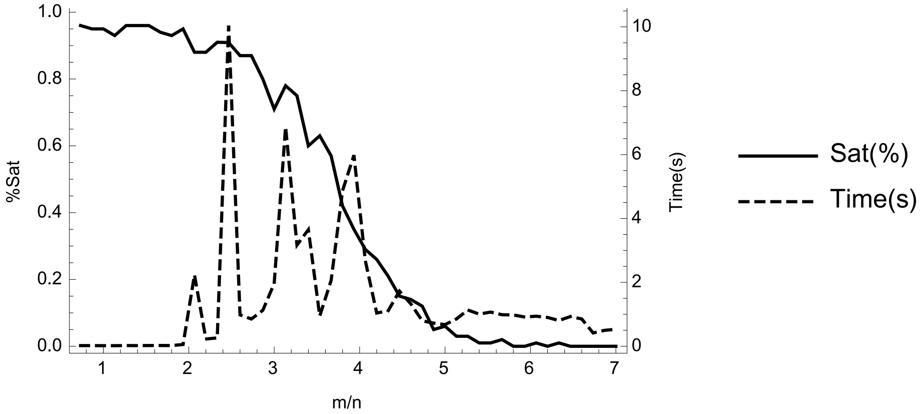
This phenomenon is characterized by the existence of a threshold value for the ratio m/n, where m is the number of clauses, and n is the number of variables, for which: for smaller values of the ratio, the SAT instances are almost certainly satisfiable and easily solved, whereas instances with larger ratio values are almost certainly unsatisfiable and also easily solved. However, with values of the ratio very closed to this threshold, the instances are, on average, very hard to solve and there is no certainty on whether the problem is satisfiable or not. As we have already noted, any 3-SAT problem can be seen as a GenPSAT instance. We tested our GenPSAT solver with random instances of 3-SAT, and observed that a phase transition occurs when the ratio m/n is about 4.3, in accordance with [62], see Figure 3.1.

Figure 3.1: Phase transition for SAT seen as a GenPSAT instance, with $n = 20$.



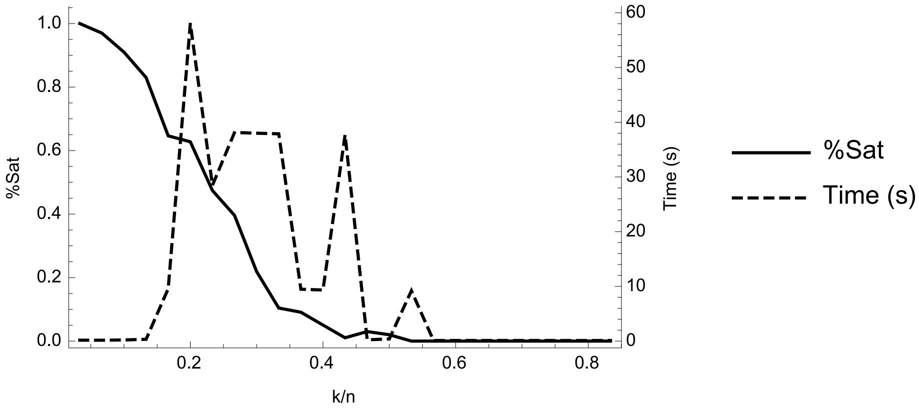
A deeper analysis of the probabilistic satisfiability problem PSAT [59, 60] has shown the presence of a phase transition behaviour for PSAT for a ratio m/n, where m is the number of clauses and n is the number of variables. We tested random PSAT instances with the number of probabilistic formulas $k = 2$, $n = 15$ and m ranging from 1 to 105 in steps of 2. For each value of m, we generated 100 PSAT instances. The obtained results are presented in Figure 3.2.

Figure 3.2: PSAT phase transition seen as a GenPSAT instance, with $n = 15$ and $k = 2$.



We highlight that the analysis of the existence of a phase transition with variation on k (instead of a variation on m) is essential for a deep understanding of the phase transition of the probabilistic satisfiability problem (instead of the phase transition of the satisfiability problem for propositional formulas in the presence of probabilistic formulas). For this purpose, we tested random PSAT instances with $n = 30$, $m = 40$ and k ranging from 1 to 25, and also observed a phase transition with respect to k/n based on 100 instances for each value of k, see Figure 3.3.

Figure 3.3: PSAT phase transition seen as a GenPSAT instance, with $n = 30$ and $m = 40$.



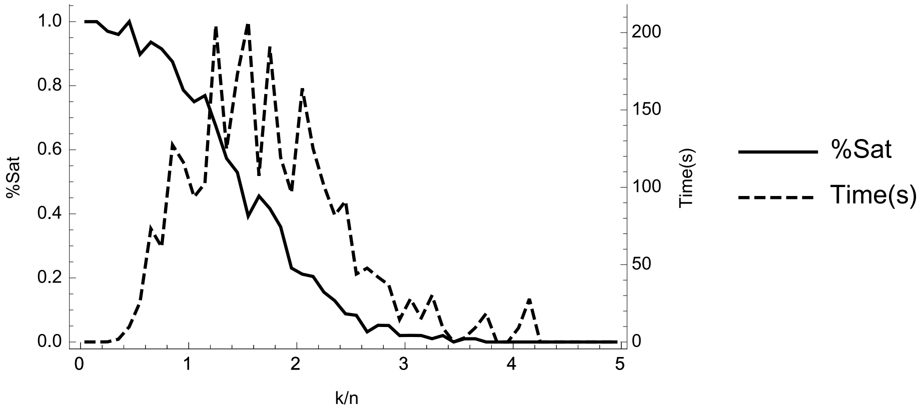
In [48], this phase transition analysis was performed on a generalization of the probabilistic satisfiability problem, GPSAT, which consists in Boolean combinations of simple probabilistic formulas.

In what concerns our generalized version of probabilistic satisfiability GenPSAT, notice that a randomly sampled probabilistic formula can easily be inconsistent by itself, e.g., when it implies one of the probabilities is greater than 1. Because of this, the sampling of the coefficients was performed in such a way that this case does not occur.

GenPSAT gives us a wider scope of ratios to study the phase transition behaviour. Due to its generalized nature, we have four dimensions to explore: the number of variables n, the number of clauses m, the number of probabilistic formulas k and the maximum size of the linear combination into the probabilistic formulas ℓ. We analyze the presence of phase transition for the ratios k/n and m/n and address the analysis of the phase transition for the variation of ℓ/n in future work.

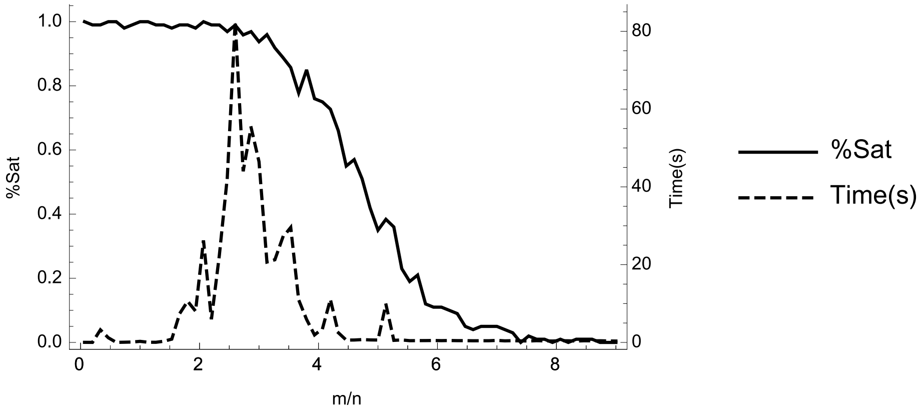
By performing random tests, we observe the presence of a phase transition for the ratio of k/n with a very short stage of satisfiable formulas. This is explained since a GenPSAT instance is more likely to be unsatisfiable. Figure 3.4 represents the phase transition for random GenPSAT instances with $n = 20$, $m = 10$ and k ranging from 1 to 100 in steps of 2. We generated 100 instances for each value of k.

Figure 3.4: Phase transition for random GenPSAT instances, with $n = 20$ and $m = 10$.



On the other hand, when the parameters n and k are fixed, we are also able to detect a phase transition. Figure 3.5 represents the result of testing random GenPSAT instances with $n = 15$, $k = 2$ and m ranging from 1 to 105 in steps of 2. For each value of m we generated 100 GenPSAT instances.

Figure 3.5: Phase transition for random GenPSAT instances, with $n = 15$ and $k = 2$.



3.5 Concluding Remarks

Throughout this chapter we explored a generalized version of probabilistic satisfiability, GenPSAT. Capitalizing on its NP-completeness, we presented a polynomial reduction from GenPSAT to MIP, which was proved to be correct. Since the translated MIP problem only suffers a quadratic growth, we were able to solve reasonably sized instances for different values of the parameters: number of variables, clauses and probabilistic formulas. Seeing that an instance can be parametrized by different combinations of these parameters, we are able to make a rich analysis of the phase transition, by analyzing the behaviour for different ratios.

The study of the phase transition taking into account also the size of the linear combination in the probabilistic formulas, as well as a 4^{th}-dimensional analysis on the variation of the parameters would also be interesting and would provide a richer setting.

We built a tool that implements this algorithm, which although being able to solve reasonably sized instances, can be greatly improved and optimized. In this sense, we explored a reduction of GenPSAT to SMT, but it has not revealed to be more efficient in general. We believe that an improvement on the performance of the GenPSAT solver would go through a reduction of the number of constraints (val1) and (val2) by strictly assigning a $[0, 1]$-variable α to the variables that occur in probability formulas, instead of doing it to all the fixed propositional variables. This would have a direct impact on the quadratic parameters that characterize the number of generated inequalities by the GenToMIP procedure. Another key factor stands on the chosen programming language.

Chapter 4

Probabilistic Logic over Equations and Domain Restrictions

Now all the components come into play: we propose and study a probabilistic logic over an algebraic basis, including equations and domain restrictions, that combines aspects from classical propositional logic and equational logic with an exogenous approach to quantitative probabilistic reasoning. This is the probabilistic logic that we envisaged from the beginning and is aimed at dealing with the kind of reasoning used in the verification of security protocols, namely in a more general analysis of offline guessing attacks [18] where the usual Dolev-Yao intruder [53] is extended with some cryptanalytic power [41, 83].

The probabilistic logic over equations and domain restrictions (DEqPrL) is designed as a *global* probabilistic logic built on top of a *local* equational base with domain restrictions. These two layers are permeated by a quantification mechanism over possible *outcomes* and a quantitative probability operator. Intuitively, we refer to algebraic terms using *names* whose concrete values are gathered in a set of possible *outcomes*, which in turn is endowed with a probability space. The local layer of DEqPrL allows us to reason about equational constraints and domain restrictions on individual outcomes. At the global layer, we can state and reason about qualitative and quantitative properties of the set of all possible outcomes. Not unexpectedly, the quantification we use can be understood as a $S5$-like modality, which also explains why we do not need to consider nested quantifiers. Arguably in the same lines, we will not consider nested probability operators [96]. DEqPrL is an extension of EqCL (Chapter 2) with probabilities and domain restrictions. Our approach bears important similarities with exogenous logics in the sense of [77], and with probabilistic logics as developed, for instance, in [58]. We present a sound and (weakly) complete axiomatization for the logic, parameterized by an equational specification of the algebraic basis coupled with the intended domain restrictions. We also show that the satisfiability problem for the logic is decidable, under the assumption that its algebraic basis is given by means of a convergent rewriting system and, additionally, that the axiomatization of domain restrictions enjoys a suitable subterm property. As a consequence, the validity problem is also decidable. Our decidability proof is actually more informative, as we develop a satisfiability algorithm for DEqPrL by means of a polynomial reduction to the Satisfiability Modulo Theories with respect to the theory of quantifier-free linear arithmetic over the integers and reals ($\mathsf{QF\_LIRA}$). This algorithm is proved to be correct. Afterwards, a solver for the satisfiability problem is tested in information security problems for verifying and estimating the probability of attacks to cryptographic protocols. Under the assumption that the rewriting system that defines the equational basis underlying the logic is also subterm convergent, we also show that the resulting satisfiability problem is NP, and thus the validity problem is coNP.

The chapter is outlined as follows: in Section 4.1 we define the syntax and semantics of DEqPrL, in Section 4.2 we provide a suitable deductive system, whose soundness and (weak) completeness we prove in Section 4.3, assuming that we are given a clausal specification of the algebraic basis and a finite axiomatization for domain restrictions; Section 4.4 is dedicated to showing, by reduction to $\mathsf{QF\_LIRA}$, that satisfiability and validity in our logic are decidable whenever the equational basis is given by means of a convergent rewriting system and the axiomatization for domain restrictions enjoys a suitable property; finally, in Section 4.5, we explore meaningful examples, including an estimation of the probability of offline guessing attacks to simple security protocols. This work was submitted for publication (see [85]); it is now presented with some reformulations, including a more detailed analysis of satisfiability and complexity results.

4.1 Syntax and Semantics

The logic DEqPrL relies on fixing a signature F, a set of variables X, and a finite set $\mathcal{D}$ of domain names. We also introduce a countable set of *names* N, distinct from algebraic variables.

We are already familiar with the designation of *algebraic* and *nominal terms* for $T(X)$ and $T(N)$, respectively. Recall that $\mathsf{vars}(t)$ stands for the set of variables occurring in $t \in T(X)$, whereas *names*(t) stands for the set of names that occur in $t \in T(N)$. When $\mathsf{names}(t) = \varnothing$, we say that $t \in T(N)$ is a *nameless term.*

The local language of the logic is designed to express equational constraints and domain restrictions; it is built on top of the sets of equations $\mathsf{Eq}(N)$ and domain restrictions $\mathsf{DRes}(N)$ defined in Section 1.3. The set Loc of local formulas is defined by the following grammar:

$$\mathsf{Loc} ::= \mathsf{Eq}(N) \mid \mathsf{DRes}(N) \mid \neg\mathsf{Loc} \mid \mathsf{Loc} \wedge \mathsf{Loc} \ .$$

Additionally, we want to express global properties of local formulas, either by quantification or by extracting probabilities. For the purpose, we need a term language Term consisting of linear probabilistic terms defined by the grammar:

$$\mathsf{Term} ::= \mathbb{Q} \cdot \mathsf{Pr}(\mathsf{Loc}) + \cdots + \mathbb{Q} \cdot \mathsf{Pr}(\mathsf{Loc}) \ \ ,$$

which we use to define the set Prob of probabilistic statements as follows:

$$\mathsf{Prob} ::= \mathsf{Term} \geq \mathbb{Q} \ \ .$$

Finally, the language of the logic consists of the following set Glob of global formulas:

$$\mathsf{Glob} ::= \forall\mathsf{Loc} \mid \mathsf{Prob} \mid \neg\mathsf{Glob} \mid \mathsf{Glob} \wedge \mathsf{Glob} \ \ .$$

Both our local and global languages are to be interpreted classically: the former over an equational basis with domain restrictions, and the later over local formulas instead of propositional symbols. Again, we abbreviate $\neg(t_1 \approx t_2)$ by $t_1 \not\approx t_2$ for any $t_1, t_2 \in T(N)$, $\neg(t \in \mathrm{D})$ by $t \notin \mathrm{D}$ for any $t \in T(N)$, $\mathrm{D} \in \mathcal{D}$, and also use the usual abbreviations: $\psi_1 \vee \psi_2$ abbr. $\neg(\neg\psi_1 \wedge \neg\psi_2)$, $\psi_1 \to \psi_2$ abbr. $\neg\psi_1 \vee \psi_2$, $\psi_1 \leftrightarrow \psi_2$ abbr. $(\psi_1 \to \psi_2) \wedge (\psi_2 \to \psi_1)$, where either $\psi_1, \psi_2 \in \mathsf{Loc}$ or $\psi_1, \psi_2 \in \mathsf{Glob}$; given $\varphi \in \mathsf{Loc}$, $\exists\varphi$ abbreviates $\neg\forall\neg\varphi$; linear probabilistic terms have the common abbreviations saying that $q \cdot (q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell))$ abbr. $(q \cdot q_1) \cdot \mathsf{Pr}(\varphi_1) + \cdots + (q \cdot q_\ell) \cdot \mathsf{Pr}(\varphi_\ell)$, $-q \cdot w$ abbr. $(-q) \cdot w$, $w_1 + w_2$ abbr. $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) + q'_1 \cdot \mathsf{Pr}(\varphi'_1) + \cdots + q'_\ell \cdot \mathsf{Pr}(\varphi'_\ell)$, whenever w_1 is of the form $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell)$ and w_2 is of the form $q'_1 \cdot \mathsf{Pr}(\varphi'_1) + \cdots + q'_\ell \cdot \mathsf{Pr}(\varphi'_\ell)$; probabilistic formulas result from the usual abbreviations $w_1 \geq w_2 + q$ abbr. $w_1 - w_2 \geq q$, $w < q$ abbr. $\neg(w \geq q)$, $w \leq q$ abbr. $-w \geq -q$, $w > q$ abbr. $-w < -q$, $w = q$ abbr. $w \leq q \wedge w \geq q$, $q_1 \leq w \leq q_2$ abbr. $w \geq q_1 \wedge w \leq q_2$, where $\ell \geq 1$, $\varphi_1, \ldots, \varphi_\ell \in \mathsf{Loc}$, $q, q_1, q_2, \ldots, q_\ell \in \mathbb{Q}, w, w_1, w_2 \in \mathsf{Term}$.

We introduce a symbol for *local true* $\top$ abbreviating $\varphi \vee \neg\varphi$ for some $\varphi \in \mathsf{Loc}$ and the *local false* $\bot$ representing $\neg\top$. We abuse notation and denote the *global true*, $\forall\top$, and *global false*, $\forall\bot$, also by $\top$ and $\bot$.

A *literal* is a global formula in $\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc} \cup \mathsf{Prob} \cup \neg\mathsf{Prob}$. We say that a global formula is in *disjunctive normal form* (DNF) if it is a disjunction of one or more conjunctions of literals; it is in *conjunctive normal form* (CNF) if it is a conjunction of one or more disjunctions of literals.

The language of the logic allows us to make qualitative and quantitative assertions over local formulas. The universal quantification of a local formula expresses the validity of the local formula in all possible situations, whereas a probabilistic statement measures the probability of satisfying local formula(s). Boolean combinations are allowed in both local and global layers. For instance, the formula $(\mathsf{Pr}(\varphi) \leq 2 \cdot \mathsf{Pr}(\psi \wedge \neg\varphi)) \wedge (\forall\neg\psi \to \forall\neg\varphi)$ should be read as: the probability of φ does not exceed twice the probability of $\psi \wedge \neg\varphi$ and, either ψ holds in some situation or else φ never holds. Note that, contrarily to the discussion carried out by Eijck and Schwarzentruber in [108], $\forall\varphi$ implies but is not intended to be equivalent to $\mathsf{Pr}(\varphi) = 1$.

Example 4.1.1. Recall, from Examples 1.3.19 and 1.3.20 presented on Subsection 1.3.5, the algebraic characterization of the sum (xor) of single bits given by the equational theory

$$\Gamma^{\text{xor}} = \{\mathsf{zero} \oplus x \approx x, \mathsf{suc}(x) \oplus y \approx x \oplus \mathsf{suc}(y), \mathsf{suc}(\mathsf{suc}(x)) \approx x\}$$

and by the set of domain restrictions

$$\Lambda^{\text{xor}} = \{\mathsf{zero} \in \mathsf{even}, (x \in \mathsf{even} \Rightarrow \mathsf{suc}(x) \in \mathsf{odd}),\ x \in \mathsf{odd} \Rightarrow \mathsf{suc}(x) \in \mathsf{even},\ x \in \mathsf{odd} \Rightarrow x \notin \mathsf{even}\},$$

once considered the set of domain names $\mathcal{D}^{\text{xor}} = \{\mathsf{even}, \mathsf{odd}\}$. Given a name $n \in N$, we want to be able to show that a statement like

$$\mathsf{Pr}(\, n \in \mathsf{even}\,) = \mathsf{Pr}(\,\mathsf{suc}(n) \in \mathsf{odd}\,) \ \wedge \ \forall(\,\mathsf{zero} \not\approx \mathsf{suc}(\mathsf{zero})\,)$$

is a theorem of the logic whose algebraic basis is axiomatized by Γ^{xor} and whose domain restrictions are given by Λ^{xor}.

△

We extend the notion of subterm to global formulas in a standard way, and abuse notation by denoting $\mathsf{subterms}(\Psi) = \bigcup_{\psi\in\Psi} \mathsf{subterms}(\psi)$, for $\Psi \subseteq \mathsf{Glob}$. Similarly, we generalize the notion of names occurring in a term to local and global formulas. The set of subformulas of either a local or a global formula ψ is defined in the usual way and is denoted by $\mathsf{subform}(\psi)$. As usual, $\mathsf{subform}(\Psi) = \bigcup_{\psi\in\Psi} \mathsf{subform}(\psi)$.

Recall that: given a nominal term $t_0 \in T(N)$, a set of names $\tilde{n} = \{n_1, \dots, n_k\} \subseteq N$ such that $\mathsf{names}(t_0) \subseteq \tilde{n}$ and $\tilde{t} = \{t_1, \dots, t_k\} \subseteq T(N)$, $[t_0]^{\tilde{n}}_{\tilde{t}}$ is the nominal term obtained by replacing each occurrence of n_i by t_i, $i \in \{1, \dots, k\}$, i.e., $[t_0]^{\tilde{n}}_{\tilde{t}} = \sigma(t_0)$ where σ is a substitution such that $\sigma(n_i) = t_i$ for each i. Analogously, given a nominal term $t_0 \in T(N)$, a set of constant symbols $\tilde{c} = \{c_1, \dots, c_k\} \subseteq \mathrm{F}_0$ and $\tilde{t} = \{t_1, \dots, t_k\} \subseteq T(N)$ we denote by $[t_0]^{\tilde{c}}_{\tilde{t}}$ the term that is obtained by replacing each occurrence of c_i by t_i, $i \in \{1, \dots, k\}$. These notions are easily extended to local and global formulas.

Names can be thought of as being associated to values that are not made explicit, and which are possibly sampled according to some probability distribution. This is why we have brought forward the designation of the possible concretizations of names as *outcomes* in EqCL (Chapter 2). Thereby, we call *outcome* to each possible concrete assignment of values to names also in DEqPrL. Given an F-algebra $\mathbb{A}$ with carrier set A, recall that we defined an outcome as a function $\rho : N \to A$. The interpretation of terms $[\![\cdot]\!]^{\rho}_{\mathbb{A}} : T_{\mathrm{F}}(N) \to A$ is defined as usual.

Definition 4.1.2. Given an algebraic domain interpretation $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$, the *satisfaction relation for local formulas*, $\Vdash^{\text{DEqPrL}}_{\mathsf{loc}}$, is defined inductively as follows:

- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} t_1 \approx t_2$ iff $[\![t_1]\!]^{\rho}_{\mathbb{A}} = [\![t_2]\!]^{\rho}_{\mathbb{A}}$;
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} t \in \mathrm{D}$ iff $[\![t]\!]^{\rho}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D})$;
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} \neg\varphi$ iff $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \nVdash^{\text{DEqPrL}}_{\mathsf{loc}} \varphi$;
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} \varphi_1 \wedge \varphi_2$ iff $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} \varphi_1$ and $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash^{\text{DEqPrL}}_{\mathsf{loc}} \varphi_2$.

We abuse notation and, instead of $\Vdash^{\text{DEqPrL}}_{\mathsf{loc}}$, we use $\Vdash_{\mathsf{loc}}$ to represent the satisfaction relation for local formulas in DEqPrL. Note that this is an extension of the analogous notion for EqCL, but now defined for an algebraic domain interpretation rather than simply for an F-algebra.

In order to interpret global formulas we need to fix an intended set of possible outcomes for names and to endow it with a probability space, which is instrumental for evaluating probabilistic statements.

Definition 4.1.3. A DEqPrL-F-*structure* is a tuple $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ where $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ is an algebraic domain interpretation, and $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$ is a probability space composed by:

- a non-empty set $\mathsf{S} \subseteq A^N$ of *possible outcomes*,
- a σ-algebra $\mathscr{A}$ containing the sets of outcomes satisfying each local formula,
$$\{\mathsf{S}^{\varphi} \mid \varphi \in \mathsf{Loc}\} \subseteq \mathscr{A}, \text{ with } \mathsf{S}^{\varphi} = \{\rho \in \mathsf{S} \mid (\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \varphi\},$$
- a probability measure μ over $\mathscr{A}$.

Definition 4.1.4. Given a DEqPrL-F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$, we define the *satisfaction relation for global formulas*, $\Vdash^{\text{DEqPrL}}$, inductively as follows:

- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \forall\varphi$ iff $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \varphi$ for every $\rho \in \mathsf{S}$,
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q$ iff $q_1 \cdot \mu(\mathsf{S}^{\varphi_1}) + \cdots + q_\ell \cdot \mu(\mathsf{S}^{\varphi_\ell}) \geq q$,
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \neg\delta$ iff $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \nVdash^{\text{DEqPrL}} \delta$,
- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \delta_1 \wedge \delta_2$ iff $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \delta_1$ and $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \delta_2$.

As usual, given $\Delta \subseteq \mathsf{Glob}$ we write $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \Delta$ if $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\text{DEqPrL}} \delta$ for each $\delta \in \Delta$.

Our logic is parameterized by a choice of intended algebraic domain interpretations.

Definition 4.1.5. Given a class $\mathcal{I}$ of algebraic domain interpretations, the *semantic consequence relation* of DEqPrL, $\vDash_{\mathcal{I}} \subseteq (\wp(\mathsf{Glob}) \times \mathsf{Glob})$, is such that $\Delta \vDash_{\mathcal{I}} \delta$ provided that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\mathrm{DEqPrL}} \delta$ whenever $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash^{\mathrm{DEqPrL}} \Delta$, for every DEqPrL-F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}$.

Note that the semantics of DEqPrL is an enrichment of the semantics of EqCL, presented in Section 2.1, that emerged to additionally deal with the probabilistic component and domain restrictions. It goes without saying that we can abuse notation and call a DEqPrL-F-structure simply as F-structure, as well as drop the superscript for the satisfaction relation for global formulas in DEqPrL.

Independence cannot in general be expressed in DEqPrL, as its language only allows for linear combinations of probabilistic terms. This could be achieved, however, without spoiling too much the nice properties of the logic, by considering coefficients taken from real closed fields, not necessarily from $\mathbb{Q}$, in the lines of the logic for reasoning about conditional probability of Fagin, Halpern and Megiddo in [58] and of the exogenous logic of Mateus, Sernadas and Sernadas presented in [77]. However, it would result in a double exponential complexity [103], which we would like to avoid. Even so, we can highlight some simple situations where one can characterize, reason about, or at least approximate the probabilistic behaviour of independent formulas, as can be seen in the following example.

Example 4.1.6. Verification of the independence of events is easily modeled within our logic: given an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$, $\varphi, \psi \in \mathsf{Loc}$ are independent if we can find $\alpha, \beta \in \mathbb{Q}$ such that $\beta \neq 0$ and $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \mathsf{Pr}(\varphi \wedge \psi) = \alpha \;\wedge\; \mathsf{Pr}(\psi) = \beta \;\wedge\; \mathsf{Pr}(\varphi) = \frac{\alpha}{\beta}$.

More importantly, we can draw some conclusions on the estimation of probabilities by knowing about the independence of some formulas. If φ and ψ are independent, we can model the expected probabilistic behaviour of both events with a finite set of properties, defined within the logic: for fixed and appropriately chosen $n, m \in \mathbb{N}$, we can introduce $n \cdot m$ conditions

$$\mathsf{Ind}^{\varphi,\psi}_{i,j}\text{:} \qquad \mathsf{Pr}(\varphi) = \tfrac{1}{i} \wedge \mathsf{Pr}(\psi) = \tfrac{1}{j} \to \mathsf{Pr}(\varphi \wedge \psi) = \tfrac{1}{i} \cdot \tfrac{1}{j}$$

for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.

As an application, let us analyze the simpler version of one-time pad encryption scheme, which consists of encrypting a secret bit by summing to it an uniformly generated key-bit. Inspired in Example 4.1.1, we consider the signature $\mathrm{F}^{\mathrm{xor}}$, the equational theory

$$\Gamma^{\mathrm{xor}} = \{\mathsf{zero} \oplus x \approx x, \mathsf{suc}(x) \oplus y \approx x \oplus \mathsf{suc}(y), \mathsf{suc}(\mathsf{suc}(x)) \approx x\},$$

the set of domain names $\mathcal{D}^{\text{xor}} = \{\mathsf{even}, \mathsf{odd}\}$ and the set of domain clauses

$$\Lambda^{\text{xor}} = \{\mathsf{zero} \in \mathsf{even}, (x \in \mathsf{even} \Rightarrow \mathsf{suc}(x) \in \mathsf{odd}),\ x \in \mathsf{odd} \Rightarrow \mathsf{suc}(x) \in \mathsf{even},\ x \in \mathsf{odd} \Rightarrow x \notin \mathsf{even}\}.$$

Let us denote by $\mathcal{I}_{(\Gamma^{\text{xor}},\Lambda^{\text{xor}})}$ the class of algebraic domain interpretations satisfying the axiomatizations Γ^{xor} and Λ^{xor}.

Now, consider a bit s, which will be kept secret as result of its encryption with a key-bit k. The described properties on the estimation of probabilities for the conjunction of independent events enable us to semantically infer that, under the hypothesis that k is uniformly generated and that bits s and k are independent,

$$\begin{aligned}\mathsf{Hyp} = \{ \quad & \mathsf{Pr}(k \approx \mathsf{zero}) = \tfrac{1}{2},\ \ \mathsf{Pr}(k \approx \mathsf{suc}(\mathsf{zero})) = \tfrac{1}{2},\ \ \mathsf{Ind}^{s,k}_{2,2},\ \ \forall(s \approx \mathsf{zero} \vee s \approx \mathsf{suc}(\mathsf{zero})),\\ & \forall(k \approx \mathsf{zero} \vee k \approx \mathsf{suc}(\mathsf{zero}))\ \ \}\end{aligned}$$

$s \oplus k$ has uniform distribution:

$$\mathsf{Hyp} \vDash_{\mathcal{I}_{(\Gamma^{\text{xor}},\Lambda^{\text{xor}})}} \left(\mathsf{Pr}(s \oplus k \approx \mathsf{zero}) = \frac{1}{2} \wedge \mathsf{Pr}(s \oplus k \approx \mathsf{suc}(\mathsf{zero})) = \frac{1}{2}\right).$$

Notice that we can generalize the properties $\mathsf{Ind}^{\varphi,\psi}_{i,j}$ estimating the probability for the conjunction of independent event by squeezing its value. For a fixed $n \in \mathbb{N}$, $q_1, \ldots, q_n \in \mathbb{Q}$ such that $q_1 < \cdots < q_n = 1$, and independent events $\varphi, \psi \in \mathsf{Loc}$,

$\widetilde{\mathsf{Ind}}^{i_1 j_1}_{i_2 j_2}$: $\quad (q_{i_1} \le \mathsf{Pr}(\varphi) \le q_{i_2} \ \wedge\ q_{j_1} \le \mathsf{Pr}(\psi) \le q_{j_2}) \to q_{i_1} \cdot q_{j_1} \le \mathsf{Pr}(\varphi \wedge \psi) \le q_{i_2} \cdot q_{j_2},$

for $i_1, i_2, j_1, j_2 \in \{1, \ldots, n\}$, would model the estimation of bounds of the probabilities for the conjunction of independent events given bounds for the individual probabilities.

△

4.2 Deductive System

In order to obtain a sound and complete deductive system for our logic, we require that the class $\mathcal{I}$ of intended interpretations is such that its algebras are axiomatized by a set Γ of Horn clauses and the corresponding interpretations for domain names are axiomatized by a finite set Λ of domain clauses of algebraic terms. We say that Γ and Λ are *compatible* if $\mathcal{I}_{(\Gamma,\Lambda)} = \{(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \mid \mathbb{A} \Vdash \Gamma \text{ and } (\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \Vdash \Lambda\} \neq \varnothing$. Whenever Γ, Λ are not compatible, the set of models is empty and the logic becomes trivial. The interesting cases are, obviously, the ones where the equational theory and the set of domain restrictions are compatible.

Eq1 $\forall(t \approx t)$

Eq2 $\forall(t_1 \approx t_2 \to t_2 \approx t_1)$

Eq3 $\forall(t_1 \approx t_2 \ \wedge\ t_2 \approx t_3 \to t_1 \approx t_3)$

Eq4 $\forall(t_1 \approx t'_1 \wedge ... \wedge t_n \approx t'_n \to f(t_1,...,t_n) \approx f(t'_1,...,t'_n))$

EqC1 $\forall((\varphi_1 \to (\varphi_2 \to \varphi_3)) \to ((\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)))$

EqC2 $\forall(\varphi_1 \to (\varphi_2 \to \varphi_1))$

EqC3 $\forall((\neg\varphi_1 \to \neg\varphi_2) \to (\varphi_2 \to \varphi_1))$

EqC4 $\forall(\varphi_1 \to ((\varphi_1 \to \varphi_2) \to \varphi_2))$

DEq $\forall((t_1 \approx t_2 \wedge t_1 \in \mathrm{D}) \to t_2 \in \mathrm{D})$

I1 $w \geq q \vee w \leq q$

I2 $w \geq q_1 \to w > q_2$, if $q_1 > q_2$

N1 $\forall(\varphi_1 \wedge \varphi_2) \leftrightarrow (\forall\varphi_1 \wedge \forall\varphi_2)$

N2 $\forall\neg\varphi \to \neg\forall\varphi$

N3 $\neg\forall\varphi \to \forall\neg\varphi$ if $names(\varphi) = \varnothing$

N4 $\forall(\varphi_1 \leftrightarrow \varphi_2) \to (\forall\varphi_1 \leftrightarrow \forall\varphi_2)$

C1 $\delta_1 \to (\delta_2 \to \delta_1)$

C2 $(\delta_1 \to (\delta_2 \to \delta_3)) \to ((\delta_1 \to \delta_2) \to (\delta_1 \to \delta_3))$

C3 $(\neg\delta_1 \to \neg\delta_2) \to (\delta_2 \to \delta_1)$

C4 $\dfrac{\delta_1 \quad \delta_1 \to \delta_2}{\delta_2}$

P1 $\mathsf{Pr}(\varphi) \geq 0$

P2 $\mathsf{Pr}(\varphi_1 \wedge \varphi_2) + \mathsf{Pr}(\varphi_1 \wedge \neg\varphi_2) - \mathsf{Pr}(\varphi_1) = 0$

P3 $\forall(\varphi_1 \to \varphi_2) \to \mathsf{Pr}(\varphi_2) \geq \mathsf{Pr}(\varphi_1)$

P4 $\mathsf{Pr}(\top) = 1$

I3 $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \leftrightarrow q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) + 0 \cdot \mathsf{Pr}(\varphi_{\ell+1}) \geq q$

I4 $((q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q) \wedge (q'_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q'_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q')) \to ((q_1 + q'_1) \cdot \mathsf{Pr}(\varphi_1) + \cdots + (q_\ell + q'_\ell) \cdot \mathsf{Pr}(\varphi_\ell) \geq q + q')$

I5 $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \to (q' \cdot q_1) \cdot \mathsf{Pr}(\varphi_1) + \cdots + (q' \cdot q_\ell) \cdot \mathsf{Pr}(\varphi_\ell) \geq (q' \cdot q)$, for any $q' > 0$

I6 $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \leftrightarrow q_{i_1} \cdot \mathsf{Pr}(\varphi_{i_1}) + \cdots + q_{i_\ell} \cdot \mathsf{Pr}(\varphi_{i_\ell}) \geq q$, for any permutation $(i_1 \cdots i_\ell)$ of $(1 \cdots \ell)$

D(Λ) $\forall(\sigma(s_1) \in \mathrm{D}_1 \wedge ... \wedge \sigma(s_{k_1}) \in \mathrm{D}_{k_1}) \to (\sigma(s'_1) \textcircled{\in} \mathrm{D}'_1 \vee ... \vee \sigma(s'_{k_2}) \textcircled{\in} \mathrm{D}'_{k_2})$

E(Γ) $\forall(\sigma(s_1) \approx \sigma(s'_1) \wedge \ldots \wedge \sigma(s_n) \approx \sigma(s'_n) \to \sigma(s) \approx \sigma(s'))$

for every $t, t_1, t_2, t_3, \ldots, t_n, t'_1, \ldots, t'_n \in T(N)$, $\varphi, \varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_\ell, \varphi_{\ell+1} \in \mathsf{Loc}$, $\delta_1, \delta_2, \delta_3 \in \mathsf{Glob}$, $\sigma \in T(N)^X$, $w \in \mathsf{Term}$, $q, q', q_1, q_2, ..., q_\ell, q'_1, \ldots, q'_\ell \in \mathbb{Q}$, $(s_1 \in \mathrm{D}_1, \ldots, s_{k_1} \in \mathrm{D}_{k_1} \Rightarrow s'_1 \textcircled{\in} \mathrm{D}'_{k_1}, \ldots, s'_{k_2} \textcircled{\in} \mathrm{D}'_{k_2}) \in \Lambda$ and $(s_1 \approx s'_1, \ldots, s_n \approx s'_n \Rightarrow s \approx s') \in \Gamma$.

Figure 4.1: The deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$.

The deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ shown in Figure 4.1 enriches the reasoning underlying $\mathcal{H}_\Gamma$ by including a number of axioms to deal with the probabilistic component and some others to incorporate and deal with domain restrictions. $\mathcal{H}_{(\Gamma,\Lambda)}$ contains a single inference rule C4, *modus ponens*. The system combines the different dimensions of this logic: axioms Eq1-Eq4 incorporate standard equational reasoning, namely reflexivity, symmetry, transitivity and congruence; EqC1-EqC4 and C1-C4 incorporate classical reasoning for the local and global layers (just note that locally, *modus ponens* becomes axiom EqC4); N1-N4 characterize the relationship between the local and global layers across the universal quantifier; DEq represents syntactically the expected relation between equations and domain restrictions; I1-I6 incorporate properties of inequalities between rational numbers; P1-P4 represent the standard properties of probabilities; axioms $\mathsf{E}(\Gamma)$ incorporate the clausal specification Γ, whereas axioms $\mathsf{D}(\Lambda)$ characterize the constraints for domains given by Λ. We define, as usual, a deducibility relation $\vdash^{\mathrm{F}}_{(\Gamma,\Lambda)}$. We drop the superscript F whenever it is clear from context.

Basic arithmetic properties, such as $0 \cdot \mathsf{Pr}(\varphi) = 0$ or $q_1 \cdot \mathsf{Pr}(\varphi) + q_2 \cdot \mathsf{Pr}(\varphi) = (q_1 + q_2) \cdot \mathsf{Pr}(\varphi)$, are deducible in $\mathcal{H}_{(\Gamma,\Lambda)}$, as well as some expected properties of the probabilistic operator, namely $\forall\varphi \to \mathsf{Pr}(\varphi) = 1$ or $\forall(\varphi_1 \leftrightarrow \varphi_2) \to \mathsf{Pr}(\varphi_1) = \mathsf{Pr}(\varphi_2)$. Moreover, notice that DEqPrL is an extension of classical logic at both the local and global layers, so we must be able to import many properties and results from classical propositional logic. As for EqCL, we are able to see that *deduction metatheorem* (MTD) holds, that every local or global formula can be written equivalently in disjunctive normal form or even that the conjunction introduction rule (Conj) and the hypothetical syllogism (HSyll) hold. The following Lemma contains some of these results and might be useful later. Just note that the normality-like axiom takes the form of theorem N.

Lemma 4.2.1. *The following properties hold:*

MTD $\Psi \cup \{\delta\} \vdash_{(\Gamma,\Lambda)} \delta'$ *if and only if* $\Psi \vdash_{(\Gamma,\Lambda)} \delta \to \delta'$

Conj $\{\delta_1, \delta_2\} \vdash_{(\Gamma,\Lambda)} \delta_1 \wedge \delta_2$

HSyll $\{\delta_1 \to \delta_2, \delta_2 \to \delta_3\} \vdash_{(\Gamma,\Lambda)} \delta_1 \to \delta_3$

DNF $\vdash_{(\Gamma,\Lambda)} \delta \leftrightarrow \bigvee_{j=1}^{m} \left(\bigwedge_{i=1}^{n_j} \delta_i^j \right)$, *for some* $\{\delta_i^j\}_{\substack{i\in\{1,\dots,n_j\} \\ j\in\{1,\dots,m\}}} \subseteq (\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc})$

N $\vdash_{(\Gamma,\Lambda)} \forall(\varphi_1 \to \varphi_2) \to (\forall\varphi_1 \to \forall\varphi_2)$

Aux1 $\vdash_{(\Gamma,\Lambda)} \forall((\varphi_1 \to \varphi_2) \leftrightarrow \neg(\varphi_1 \wedge \neg\varphi_2))$

Aux2 $\vdash_{(\Gamma,\Lambda)} \delta_1 \to (\delta_2 \to (\delta_1 \wedge \delta_2))$

Aux3 $\vdash_{(\Gamma,\Lambda)} \forall((\neg(\varphi_1 \wedge \neg\varphi_2) \wedge \varphi_1) \leftrightarrow \varphi_1 \wedge \varphi_2)$

Aux4 $\vdash_{(\Gamma,\Lambda)} (\forall\varphi_1 \wedge \forall\varphi_2) \to \forall\varphi_2$

Aux5 $\vdash_{(\Gamma,\Lambda)} (\delta_1 \leftrightarrow \delta_2) \to (\neg\delta_1 \leftrightarrow \neg\delta_2)$

Aux6 $\vdash_{(\Gamma,\Lambda)} (\delta_1 \leftrightarrow \delta_2) \to (\delta_1 \to \delta_2)$

Aux7 $\vdash_{(\Gamma,\Lambda)} ((\delta_1 \wedge \neg\delta_3) \to \neg\delta_2) \to ((\delta_1 \wedge \delta_2) \to \delta_3)$

Aux8 $\vdash_{(\Gamma,\Lambda)} ((\delta_1 \leftrightarrow \delta_2) \wedge (\delta_2 \leftrightarrow \delta_3)) \rightarrow (\delta_1 \leftrightarrow \delta_3)$

Aux9 $\vdash_{(\Gamma,\Lambda)} (\forall(\varphi \rightarrow \varphi_1) \wedge \forall(\varphi \rightarrow \varphi_2)) \rightarrow \forall(\varphi \rightarrow (\varphi_1 \wedge \varphi_2))$

Aux10 $\vdash_{(\Gamma,\Lambda)} (\forall(\varphi_1 \rightarrow \varphi_2) \wedge \forall(\varphi_2 \rightarrow \varphi_3)) \rightarrow \forall(\varphi_1 \rightarrow \varphi_3)$

PAux1 $\vdash_{(\Gamma,\Lambda)} \forall(\varphi_1 \leftrightarrow \varphi_2) \rightarrow \mathsf{Pr}(\varphi_1) = \mathsf{Pr}(\varphi_2)$

PAux2 $\vdash_{(\Gamma,\Lambda)} \forall\varphi \rightarrow \mathsf{Pr}(\varphi) = 1$

PAux3 $\vdash_{(\Gamma,\Lambda)} \mathsf{Pr}(\varphi) = \mathsf{Pr}(\varphi)$

PAux4 $\vdash_{(\Gamma,\Lambda)} q_1 \cdot \mathsf{Pr}(\varphi) + q_2 \cdot \mathsf{Pr}(\varphi) = (q_1 + q_2) \cdot \mathsf{Pr}(\varphi)$

PAux5 $\vdash_{(\Gamma,\Lambda)} 0 \cdot \mathsf{Pr}(\varphi) = 0$

PAux6 $\vdash_{(\Gamma,\Lambda)} w = q \rightarrow w + w_1 = q + w_1$

PAux7 $\vdash_{(\Gamma,\Lambda)} \mathsf{Pr}(\bot) = 0$

where $\varphi, \varphi_1, \varphi_2, \varphi_3 \in \mathsf{Loc}$*,* $\Psi \subseteq \mathsf{Glob}$*,* $\delta, \delta', \delta_1, \delta_2, \delta_3 \in \mathsf{Glob}$*,* $w, w_1 \in \mathsf{Term}$*.*

Proof. As we stressed in the proof of Lemma 2.2.1, the deductions of the former properties look similar to the corresponding properties in in the classical context. Some of them can be found in the proof of Lemma 2.2.1.

Notice that PAux1 is immediate from P3; PAux2 and PAux3 follow from PAux1; PAux6 is immediate from PAux4 and PAux5; whereas PAux7 is easily deduced from P2, P4 and PAux6. We illustrate the deduction of PAux4 and PAux5.

PAux4 To deduce the equality we sketch the proof of both inequalities and the result follows by Conj. Assume that $q_1, q_2 > 0$. If it is not the case, proceed with the symmetric and use the abbreviations to revert the inequalities.

s_1. $\mathsf{Pr}(\varphi) = \mathsf{Pr}(\varphi)$ (instance of PAux3)

s_2. $\mathsf{Pr}(\varphi) = \mathsf{Pr}(\varphi) \rightarrow \mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi)$ (abbr.)

s_3. $\mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi)$ (apply C4 to s_1 and s_2)

s_4. $\mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi) \rightarrow q_1 \cdot \mathsf{Pr}(\varphi) \geq q_1 \cdot \mathsf{Pr}(\varphi)$ (instance of I5)

s_5. $q_1 \cdot \mathsf{Pr}(\varphi) \geq q_1 \cdot \mathsf{Pr}(\varphi)$ (apply C4 to s_3 and s_4)

s_6. $q_1 \cdot \mathsf{Pr}(\varphi) \geq q_1 \cdot \mathsf{Pr}(\varphi) \rightarrow q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) \geq 0$ (abbr.)

s_7. $q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply C4 to s_5 and s_6)

s_8. $q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) \geq 0 \rightarrow q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (instance of I3)

s_9. $q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply C4 to s_7 and s_8)

s_{10}. $q_2 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (repeat s_4-s_9 for $q_2 > 0$)

s_{11}. $q_2 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0 \rightarrow q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0$ (using I6)

s_{12}. $q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_{10} and s_{11})

s_{13}. $((q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0) \land (q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0)) \to$
$\to ((q_1 + q_2) \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0)$ (instance of $\mathsf{I4}$)

s_{14}. $(q_1 \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0) \land (q_2 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0)$ (apply Conj to s_9 and s_{12})

s_{15}. $(q_1 + q_2) \cdot \mathsf{Pr}(\varphi) - q_1 \cdot \mathsf{Pr}(\varphi) - q_2 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_{14} and s_{13})

The deduction for the other inequality is exactly the same, just noting that in s_2 the abbreviation that matters is that $-\mathsf{Pr}(\varphi) \geq -\mathsf{Pr}(\varphi)$.

PAux5 We present a sketch of the deduction of $0 \cdot \mathsf{Pr}(\varphi) = 0$.

s_1. $\mathsf{Pr}(\varphi) = \mathsf{Pr}(\varphi)$ (instance of $\mathsf{PAux3}$)

s_2. $\mathsf{Pr}(\varphi) = \mathsf{Pr}(\varphi) \to \mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi)$ (abbr.)

s_3. $\mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi)$ (apply $\mathsf{C4}$ to s_1 and s_2)

s_4. $\mathsf{Pr}(\varphi) \geq \mathsf{Pr}(\varphi) \to \mathsf{Pr}(\varphi) - \mathsf{Pr}(\varphi) \geq 0$ (abbr.)

s_5. $\mathsf{Pr}(\varphi) - \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_3 and s_4)

s_6. $\mathsf{Pr}(\varphi) - \mathsf{Pr}(\varphi) \geq 0 \to -\mathsf{Pr}(\varphi) + \mathsf{Pr}(\varphi) \geq 0$ (using $\mathsf{I6}$)

s_7. $-\mathsf{Pr}(\varphi) + \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_5 and s_6)

s_8. $((\mathsf{Pr}(\varphi) - \mathsf{Pr}(\varphi) \geq 0) \land (-\mathsf{Pr}(\varphi) + \mathsf{Pr}(\varphi) \geq 0)) \to (0 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0)$ (instance of $\mathsf{I4}$)

s_9. $(\mathsf{Pr}(\varphi) - \mathsf{Pr}(\varphi) \geq 0) \land (-\mathsf{Pr}(\varphi) + \mathsf{Pr}(\varphi) \geq 0)$ (apply Conj to s_5 and s_7)

s_{10}. $0 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_9 and s_8)

s_{11}. $0 \cdot \mathsf{Pr}(\varphi) + 0 \cdot \mathsf{Pr}(\varphi) \geq 0 \to 0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (using $\mathsf{I3}$)

s_{12}. $0 \cdot \mathsf{Pr}(\varphi) \geq 0$ (apply $\mathsf{C4}$ to s_{10} and s_{11})

For the other inequality notice that from s_{12}, we can obtain, by abbreviation that $-0 \geq -0 \cdot \mathsf{Pr}(\varphi)$, which is obviously the other inequality: $0 \geq 0 \cdot \mathsf{Pr}(\varphi)$. □

Example 4.2.2. As we have already seen, a standard example of an equational theory used in information security for formalizing (part of) the capabilities of the Dolev-Yao attacker consists in taking a signature F^{DY} with $\{\cdot\}_{\cdot}, \{\cdot\}_{\cdot}^{-1} \in \mathrm{F}_2^{\mathsf{DY}}$ representing symmetric encryption and decryption of a message with a key, $\{\!|\cdot|\!\}_{\cdot}, \{\!|\cdot|\!\}_{\cdot}^{-1} \in \mathrm{F}_2^{\mathsf{DY}}$ now representing asymmetric encryption of a message with a public key or decryption with a private key, $\mathsf{pk}(\cdot), \mathsf{prv}(\cdot) \in \mathrm{F}_1^{\mathsf{DY}}$ representing public and private keys for a principal, $(\cdot,\cdot) \in \mathrm{F}_2^{\mathsf{DY}}$ representing message pairing, and $\pi_1, \pi_2 \in \mathrm{F}_1^{\mathsf{DY}}$ representing projections. The equational properties of these operations can be axiomatized by the subterm theory:

$$\Gamma^{\mathsf{DY}} = \{\{\{x_1\}_{x_2}\}_{x_2}^{-1} \approx x_1, \{\!|\{\!|x_1|\!\}_{\mathsf{pk}(x_2)}|\!\}_{\mathsf{prv}(x_2)}^{-1} \approx x_1, \pi_1(x_1, x_2) \approx x_1, \pi_2(x_1, x_2) \approx x_2\}.$$

Considering a suitable set of domain names, for instance we may take

$$\mathcal{D}^{\mathsf{DY}} = \{\mathsf{sym\_key}, \mathsf{pub\_key}, \mathsf{priv\_key}, \mathsf{principals}, \mathsf{plaintext}, \mathsf{ciphertext}, \mathsf{conc}\},$$

we can also impose some usual domain restrictions:

$$\begin{aligned}\Lambda^{\mathsf{DY}} = \{ \quad & (k \in \mathsf{sym\_key}, t \in \mathsf{plaintext} \Rightarrow \{t\}_k \in \mathsf{ciphertext}),\\ & (k \in \mathsf{sym\_key}, t \in \mathsf{ciphertext} \Rightarrow \{t\}_k^{-1} \in \mathsf{plaintext}),\\ & (n \in \mathsf{principals} \Rightarrow \mathsf{pk}(n) \in \mathsf{pub\_key}), \ \ (n \in \mathsf{principals} \Rightarrow \mathsf{prv}(n) \in \mathsf{priv\_key}),\\ & (t \in \mathsf{plaintext}, k \in \mathsf{pub\_key} \Rightarrow \{\!|t|\!\}_k \in \mathsf{ciphertext}),\\ & (t \in \mathsf{ciphertext}, k \in \mathsf{priv\_key} \Rightarrow \{\!|t|\!\}_k^{-1} \in \mathsf{plaintext}),\\ & (t \in \mathsf{plaintext}, t' \in \mathsf{plaintext} \Rightarrow (t, t') \in \mathsf{conc}), \ \ (t \in \mathsf{conc} \Rightarrow t \in \mathsf{plaintext}),\\ & (t \in \mathsf{conc} \Rightarrow \pi_1(t) \in \mathsf{plaintext}), \ \ (t \in \mathsf{conc} \Rightarrow \pi_2(t) \in \mathsf{plaintext}) \ \ \} \ .\end{aligned}$$

The first domain restriction, for instance, is intended to mean that the encryption of a plaintext with a symmetric key should always lead to a ciphertext. As a result, we can deduce from DEqPrL conditions to rule out the possibility of an attack, like

$$\forall (k \in \mathsf{sym\_key} \wedge m \in \mathsf{plaintext}) \vdash_{(\Gamma^{\mathsf{DY}}, \Lambda^{\mathsf{DY}})} \forall (\{\{m\}_k\}_{k^*}^{-1} \notin \mathsf{plaintext} \rightarrow k \not\approx k^*),$$

which states that whenever an attempt to guess the secret key k leads to a message outside the scope of plaintexts, the value of k has certainly not been guessed correctly. We can also deduce a bound for the probability of an attack to the symmetric scheme:

$$\mathsf{Pr}(k \approx k^*) = q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) \vdash_{(\Gamma^{\mathsf{DY}}, \Lambda^{\mathsf{DY}})} \forall (k^* \in \mathsf{sym\_key}) \rightarrow \mathsf{Pr}(\{\{m\}_k\}_{k^*}^{-1} \approx m) \geq q, \quad (4.1)$$

asserting that even assuming that a guess k^* to the secret key k is indeed a symmetric key, guessing its concrete value is not simpler than decrypting a message encrypted with k. A deduction of (4.1) follows from a proof (see the sketch below) of:

$$\mathsf{Pr}(k \approx k^*) = q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}), \forall (k^* \in \mathsf{sym\_key}) \vdash_{(\Gamma^{\mathsf{DY}}, \Lambda^{\mathsf{DY}})} \mathsf{Pr}(\{\{m\}_k\}_{k^*}^{-1} \approx m) \geq q,$$

follows by the use of MTD.

s_1. $\mathsf{Pr}(k \approx k^*) = q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key})$ (hypothesis)
s_2. $\forall (k^* \in \mathsf{sym\_key})$ (hypothesis)
s_3. $\forall (k^* \in \mathsf{sym\_key}) \rightarrow \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 1$ (PAux2)
s_4. $\mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 1$ (C4($\mathsf{s}_2, \mathsf{s}_3$))
s_5. $\mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 1 \rightarrow q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = q$ (I5)
s_6. $q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = q$ (C4($\mathsf{s}_4, \mathsf{s}_5$))
s_7. $q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = q \rightarrow \mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = \mathsf{Pr}(k \approx k^*) - q$ (PAux6)
s_8. $\mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = \mathsf{Pr}(k \approx k^*) - q$ (C4($\mathsf{s}_6, \mathsf{s}_7$))
s_9. $\mathsf{Pr}(k \approx k^*) = q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) \rightarrow \mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 0$ (abbr.)
s_{10}. $\mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 0$ (C4($\mathsf{s}_9, \mathsf{s}_{10}$))
s_{11}. $(\mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 0 \wedge \mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = \mathsf{Pr}(k \approx k^*) - q) \rightarrow$
$\rightarrow \mathsf{Pr}(k \approx k^*) - q = 0$ (PAux6)
s_{12}. $\mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = 0 \wedge \mathsf{Pr}(k \approx k^*) - q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) = \mathsf{Pr}(k \approx k^*) - q$ (Conj($\mathsf{s}_{10}, \mathsf{s}_8$))
s_{13}. $\mathsf{Pr}(k \approx k^*) - q = 0$ (C4(s_{12}, s_{11}))
s_{14}. $\mathsf{Pr}(k \approx k^*) - q = 0 \rightarrow \mathsf{Pr}(k \approx k^*) = q$ (abbr.)
s_{15}. $\mathsf{Pr}(k \approx k^*) = q$ (C4($\mathsf{s}_{13}, \mathsf{s}_{14}$))
s_{16}. $\forall (\{\{m\}_k\}_k^{-1} \approx m)$ (E(Γ^{DY}))
s_{17}. $\forall (\{\{m\}_k\}_k^{-1} \approx m) \rightarrow \forall (k \approx k^* \rightarrow \{\{m\}_k\}_k^{-1} \approx m)$ (EqC2 + N)
s_{18}. $\forall (k \approx k^* \rightarrow \{\{m\}_k\}_k^{-1} \approx m)$ (C4($\mathsf{s}_{22}, \mathsf{s}_{23}$))

s_{19}. $\forall(k \approx k^* \to \{\{m\}_k\}^{-1}_{k^*} \approx \{\{m\}_k\}^{-1}_{k})$ (Eq4)

s_{20}. $(\forall(k \approx k^* \to \{\{m\}_k\}^{-1}_{k} \approx m) \wedge \forall(k \approx k^* \to \{\{m\}_k\}^{-1}_{k^*} \approx \{\{m\}_k\}^{-1}_{k})) \to$
$\to \forall(k \approx k^* \to (\{\{m\}_k\}^{-1}_{k} \approx m \wedge \{\{m\}_k\}^{-1}_{k^*} \approx \{\{m\}_k\}^{-1}_{k}))$ (Aux9)

s_{21}. $\forall(k \approx k^* \to (\{\{m\}_k\}^{-1}_{k} \approx m \wedge \{\{m\}_k\}^{-1}_{k^*} \approx \{\{m\}_k\}^{-1}_{k}))$ $(\mathsf{C4}(\mathsf{s}_{19}, \mathsf{s}_{20}))$

s_{22}. $\forall((\{\{m\}_k\}^{-1}_{k^*} \approx \{\{m\}_k\}^{-1}_{k} \wedge \{\{m\}_k\}^{-1}_{k} \approx m) \to \{\{m\}_k\}^{-1}_{k^*} \approx m)$ (Eq3)

s_{23}. $\forall(k \approx k^* \to \{\{m\}_k\}^{-1}_{k^*} \approx m)$ (using Aux10, s_{21}, s_{22})

s_{24}. $\forall(k \approx k^* \to \{\{m\}_k\}^{-1}_{k^*} \approx m) \to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq \mathsf{Pr}(k \approx k^*)$ (P3)

s_{25}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq \mathsf{Pr}(k \approx k^*)$ $(\mathsf{C4}(\mathsf{s}_{23}, \mathsf{s}_{24}))$

s_{26}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq \mathsf{Pr}(k \approx k^*) \to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) \geq 0$ (I4)

s_{27}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) \geq 0$ $(\mathsf{C4}(\mathsf{s}_{25}, \mathsf{s}_{26}))$

s_{28}. $\mathsf{Pr}(k \approx k^*) = q \to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) = \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q$ (PAux6)

s_{29}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) = \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q$ $(\mathsf{C4}(\mathsf{s}_{15}, \mathsf{s}_{28}))$

s_{30}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) = \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q \to$
$\to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q \geq \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*)$ (abbr.)

s_{31}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q \geq \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*)$ $(\mathsf{C4}(\mathsf{s}_{29}, \mathsf{s}_{30}))$

s_{32}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q \geq \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) \to$
$\to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q - \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) + \mathsf{Pr}(k \approx k^*) \geq 0$ (abbr.)

s_{33}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q - \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) + \mathsf{Pr}(k \approx k^*) \geq 0$ $(\mathsf{C4}(\mathsf{s}_{31}, \mathsf{s}_{32}))$

s_{34}. $(\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - \mathsf{Pr}(k \approx k^*) \geq 0 \wedge \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) - q - \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) + \mathsf{Pr}(k \approx k^*) \geq 0) \to$
$\to \mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq q$ (I3, I4)

s_{35}. $\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq q$ $(\mathsf{C4}(\mathsf{Conj}(\mathsf{s}_{27}, \mathsf{s}_{33}), \mathsf{s}_{34}))$

$\triangle$

4.3 Soundness and Completeness

We now show that $\mathcal{H}_{(\Gamma,\Lambda)}$ is a sound and weakly complete proof system for the logic based on the class $\mathcal{I}_{(\Gamma,\Lambda)}$ of algebraic domain interpretations. In this way, we ensure that the theorems deduced from $\mathcal{H}_\Gamma$ correspond exactly to the valid formulas entailed from the semantics that we have set out.

Theorem 4.3.1. *The deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ is sound.*

Proof. The proof of soundness follows by induction on the structure of the proofs. It is straightforward to check the soundness of each axiom and deduction rule against our semantics, however we detail some of them, due to their different nature.

DEq To check that DEq is valid, consider an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}_{(\Gamma,\Lambda)}$, $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$, and let $\rho \in \mathsf{S}$ represent any possible outcome for which $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} t_1 \approx t_2$ and $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} t_1 \in \mathrm{D}$, i.e., $[\![t_1]\!]^{\rho}_{\mathbb{A}} = [\![t_2]\!]^{\rho}_{\mathbb{A}}$ and $[\![t_1]\!]^{\rho}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D})$. We immediately conclude that $[\![t_2]\!]^{\rho}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D})$.

P3 For P3 let us consider an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}_{(\Gamma,\Lambda)}$, $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$, such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall(\varphi_1 \to \varphi_2)$. This means that every single outcome satisfying φ_1 also satisfies φ_2 or, in other words, $\mathsf{S}^{\varphi_1} \subseteq \mathsf{S}^{\varphi_2}$. Since μ is a probability measure, it then follows that $\mu(\mathsf{S}^{\varphi_1}) \leq \mu(\mathsf{S}^{\varphi_2})$, i.e., $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \mathsf{Pr}(\varphi_1) \leq \mathsf{Pr}(\varphi_2)$.

Axioms of inequalities follow from the properties of addition and multiplication on the rational numbers. Hence, we move forward to prove the validity of $\mathsf{D}(\Lambda)$

$\mathsf{D}(\Lambda)$ For the validity of $\mathsf{D}(\Lambda)$, let us consider a substitution $\sigma \in T(N)^X$, a domain clause $(t_1 \in \mathrm{D}_1, \ldots, t_{k_1} \in \mathrm{D}_{k_1} \Rightarrow t'_1 \circledcirc \mathrm{D}'_1, \ldots, t'_{k_2} \circledcirc \mathrm{D}'_{k_2}) \in \Lambda$ and an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}_{(\Gamma,\Lambda)}$ and $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$. We want to check that

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall(\sigma(t_1) \in \mathrm{D}_1 \wedge \ldots \wedge \sigma(t_{k_1}) \in \mathrm{D}_{k_1} \rightarrow \sigma(t'_1) \circledcirc \mathrm{D}'_1 \vee \ldots \vee \sigma(t'_{k_2}) \circledcirc \mathrm{D}'_{k_2})$$

so, let $\rho \in A^N$ be any outcome in S and assume that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \sigma(t_1) \in \mathrm{D}_1 \wedge \ldots \wedge \sigma(t_{k_1}) \in \mathrm{D}_{k_1}$. Since the algebraic domain interpretation $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ satisfies all the domain clauses in Λ provided that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}) \in \mathcal{I}_{(\Gamma,\Lambda)}$, we only need to remark that $[\![\cdot]\!]^{\rho}_{\mathbb{A}} \circ \sigma : X \to A$ to conclude that, indeed, $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \sigma(t'_1) \circledcirc \mathrm{D}'_1 \vee \ldots \vee \sigma(t'_{k_2}) \circledcirc \mathrm{D}'_{k_2}$. □

In contrast to EqCL, the introduction of probabilistic terms over the rationals carries the expected cost of losing the strong version of completeness (see, for instance, [58, 77]). Clearly, our semantic consequence relation is not compact as we have that $\{w \leq \frac{1}{n} \mid n \in \mathbb{N}\} \vDash_{(\Gamma,\Lambda)} w \leq 0$, but $\Delta \nvDash_{(\Gamma,\Lambda)} w \leq 0$ for any finite set $\Delta \subset \{w \leq \frac{1}{n} \mid n \in \mathbb{N}\}$, which implies that our (finitary) deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ cannot aim at strong completeness.

Theorem 4.3.2. *The deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ is weakly complete.*

Proof. As usual, the proof of completeness follows by contraposition and consists in finding a model for the negation of an unprovable formula. Hence, we assume that $\nvdash_{(\Gamma,\Lambda)} \delta$ and build an F-structure satisfying $\neg\delta$. The construction combines several known techniques from equational logic, first-order logic and probabilistic logic, which interact in a non-trivial way. It extends the approach presented in the proof of completeness of EqCL (Theorem 2.3.3) to further deal with probabilities and domain restrictions.

Once assumed that $\nvdash_{(\Gamma,\Lambda)} \delta$, let us begin by writing the consistent formula $\neg\delta$ in disjunctive normal form as $\psi_1 \vee \cdots \vee \psi_m$. Then, we choose a consistent disjunct ψ_j, of the form

$$\psi_j^1 \wedge \cdots \wedge \psi_j^{n_j}, \tag{4.2}$$

and define $\mathsf{RelF} = \{\psi_j^1, \ldots, \psi_j^{n_j}\} \subseteq \mathsf{Glob}$ to be the set of *relevant formulas* that should be satisfied in the final F-structure. Again, a Henkin construction [69] will enable us to define the F-algebra we are looking for. For this purpose, we add to the signature a new constant $c_{\varphi,n}$ for each $\varphi \in \mathsf{Loc}$ and $n \in N$, obtaining a signature $\mathrm{F}^+ = \{\mathrm{F}^+_n\}_{n\in\mathbb{N}}$ coinciding with F in all but

$$\mathrm{F}^+_0 = \mathrm{F}_0 \cup \left(\bigcup_{\varphi\in\mathsf{Loc}} \{c_{\varphi,n_0} \mid n_0 \in N\} \right).$$

Afterwards, we fix an enumeration for $\mathsf{Loc} \times \mathsf{Loc}$ and further extend the set RelF with witnesses for negated global formulas and with suitable certifications for non-negative global formulas, through the following inductive definition:

$$\begin{array}{lcll} W_0 & = & \mathsf{RelF} & \\ W_{i+1} & = & W_i \cup \left\{\neg\forall\varphi_i^1 \to \left(\forall[\neg\varphi_i^1]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}} \wedge \left(\forall\varphi_i^2 \to \forall[\varphi_i^2]^{\tilde{n}}_{\tilde{c}_{\varphi_i^1}}\right)\right)\right\} & \text{for each } i \in \mathbb{N}, \end{array}$$

where $\mathsf{names}(\varphi_i^1) \cup \mathsf{names}(\varphi_i^2) = \tilde{n} = \{n_1, \ldots, n_k\}$, $\tilde{c}_\varphi = \{c_{\varphi,n_1}, \ldots, c_{\varphi,n_k}\}$.

Then consider the consistent set $W = \bigcup_{i\in\mathbb{N}} W_i \subseteq \mathsf{Glob}_{F^+}$ (see Lemma 2.3.5 for details), and fix a maximal consistent extension Ξ of W, whose existence is guaranteed by Lindenbaum's Lemma. Then consider the F^+-algebra $\mathbb{A} = \mathbb{T}_{\mathrm{F}^+}(\varnothing)_{/\equiv}$, where the congruence relation $\equiv \subseteq (\mathbb{T}_{\mathrm{F}^+}(\varnothing) \times \mathbb{T}_{\mathrm{F}^+}(\varnothing))$ is given by:

$$t_1 \equiv t_2 \text{ if } \forall(t_1 \approx t_2) \in \Xi \ .$$

The relation $\equiv$ is a congruence as consequence of axioms $\mathsf{Eq1}$-$\mathsf{Eq4}$ and theorem N. A domain interpretation is then taken accordingly to the aforementioned maximal consistent set Ξ:

$$\mathrm{I}^{\mathbb{A}}(\mathrm{D}) = \{[t]_\equiv \mid \forall(t \in \mathrm{D}) \in \Xi \text{ and } t \in T_{\mathrm{F}^+}(\varnothing)\} \text{ for each } \mathrm{D} \in \mathcal{D}.$$

- $\mathbb{A}$ **satisfies** Γ**:** by definition of $\equiv$, $\mathsf{E}(\Gamma)$, $\mathsf{C4}$, N, and recalling that Ξ is a maximal consistent set, it is easy to check that $\mathbb{A} \Vdash \Gamma$.

- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ **verifies** Λ**:** given $\left(t_1 \in \mathrm{D}_1, \ldots, t_{k_1} \in \mathrm{D}_{k_1} \to t'_1 \textcircled{\in} \mathrm{D}'_1, \ldots, t'_{k_2} \textcircled{\in} \mathrm{D}_{k_2}\right) \in \Lambda$ and $\varsigma \in A^X$, notice that ς results from applying a substitution $\sigma \in T_{\mathrm{F}^+}(\varnothing)^X$ and then a quotient by $\equiv$. Assume that $[\![t_i]\!]^\varsigma_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}_i)$ for each $1 \leq i \leq k_i$, which means that for each $1 \leq i \leq k_i$, $[\sigma(t_i)]_\equiv \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}_i)$ or, equivalently, $\forall(\sigma(t_i) \in \mathrm{D}_i) \in \Xi$. It means that $\forall(\sigma(t_1) \in \mathrm{D}_1 \wedge \cdots \wedge \sigma(t_{k_1}) \in \mathrm{D}_{k_1}) \in \Xi$ and, from $\mathsf{D}(\Lambda)$, it follows that $\forall(\sigma(t'_1) \textcircled{\in} \mathrm{D}'_1 \vee \cdots \vee \sigma(t'_{k_2}) \textcircled{\in} \mathrm{D}'_{k_2}) \in \Xi$. But Ξ is maximal consistent with respect to the deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ and $\sigma(t'_1), \ldots, \sigma(t'_{k_2})$ are nameless terms, so it follows that exists $j \in \{1, \ldots, k_2\}$ such that $\forall(\sigma(t'_j) \textcircled{\in} \mathrm{D}'_j) \in \Xi$.

We note that each negated global formula in the maximal consistent set, $\neg\forall\varphi \in \Xi$, leads to an outcome assigning each name to the equivalence class of the appropriate constant:

$$\begin{array}{rccl} \rho^{\neg\forall\varphi}: & N & \to & A \\ & n & \mapsto & [c_{\varphi,n}]_\equiv . \end{array}$$

Thus, consider the set of possible outcomes $\mathsf{S} = \left\{\rho^{\neg\forall\varphi} \mid \neg\forall\varphi \in \Xi\right\}$. Note that $\mathsf{S} \neq \varnothing$ since the conjugation of the reflexivity axiom $\mathsf{Eq1}$ with the axiom that enables the negation to be passed through the universal quantifier, $\mathsf{N2}$, implies that $\neg\forall(t \not\approx t) \in \Xi$, for each $t \in T(N)$.

In order to define a probability space, we can assume without loss of generality that there exists at least one probabilistic relevant formula in (4.2), $\mathsf{RelF} \cap (\mathsf{Prob} \cup \neg\mathsf{Prob}) \neq \varnothing$. Just to make sure, note that otherwise we can simply assign a degenerated probability distribution, assigning probability 1 to some outcome in S and 0 to the others.

That said, a probability space is then defined, in the lines of [58], and starts by choosing carefully a set of atoms of interest: initially we collect in Ω all the local formulas occurring inside probabilistic formulas of RelF,

$$\Omega = \bigcup_{\psi \in \mathsf{RelF} \cap (\mathsf{Prob} \cup \neg\mathsf{Prob})} \mathsf{InPr}(\psi), \text{ where}$$

$\mathsf{InPr}(q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq b) = \mathsf{InPr}(\neg(q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq b)) = \{\varphi_1, \ldots, \varphi_\ell\}$, and then use it to define the suitable atoms $\Theta = \left\{ \bigwedge_{\gamma \in \Upsilon} \gamma \wedge \bigwedge_{\omega \in \Omega \smallsetminus \Upsilon} \neg\omega \mid \Upsilon \subseteq \Omega \right\}$. We consider a representative outcome for each element of $\theta \in \Theta$, whenever it is possible: if $\mathsf{S}^\theta \neq \varnothing$, choose $\rho_\theta \in \mathsf{S}^\theta$ and let us represent the probability assigned to ρ_θ by x_θ; otherwise, if $\mathsf{S}^\theta = \varnothing$, i.e. $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall\neg\theta$, fix $x_\theta = 0$. The accuracy of Θ immediately implies that $\bigcup_{\theta \in \Theta} \mathsf{S}^\theta = \mathsf{S}$ and $\mathsf{S}^{\theta_1} \cap \mathsf{S}^{\theta_2} = \varnothing$, for each $\theta_1 \neq \theta_2$.

The set Θ has the crucial local formulas to define the system of inequalities

$$\begin{cases} q_1 \cdot x_{\varphi_1} + \cdots + q_\ell \cdot x_{\varphi_\ell} \geq q, & \text{for each } q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \in \mathsf{RelF} \\ q_1 \cdot x_{\varphi_1} + \cdots + q_\ell \cdot x_{\varphi_\ell} < q, & \text{for each } q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) < q \in \mathsf{RelF} \\ \sum\limits_{\theta \in \Theta \text{ st } \theta \to \varphi} x_\theta = x_\varphi, & \text{for each } \varphi \in \Omega \\ \sum\limits_{\theta \in \Theta} x_\theta = 1 & \\ x_\theta = 0, & \text{for each } \theta \in \Theta \text{ such that } \mathsf{S}^\theta = \varnothing \\ x_\theta \geq 0, & \text{for all } \theta \in \Theta \end{cases} \tag{4.3}$$

We claim that this system of inequalities has a solution. Indeed, using Fagin, Halpern and Megiddo's result of soundness and completeness for the axioms of inequality (see Section 4 of [58]), we know that (4.3) is unsatisfiable if and only if it is inconsistent. But it leads to a contradiction, as we can find a global formula that represents this system of inequalities within DEqPrL. Let us look at this in more detail!

To write down a global formula that represents the system (4.3), let us fix an order on elements of Ω: $\Omega = \{\varphi_1, \ldots, \varphi_{|\Omega|}\}$. Then, consider the successive application of axiom P2 to deduce that

$$\begin{aligned} \mathsf{Pr}(\varphi_1) &= \mathsf{Pr}(\varphi_1 \wedge \varphi_2) + \mathsf{Pr}(\varphi_1 \wedge \neg\varphi_2) = \\ &= \mathsf{Pr}(\varphi_1 \wedge \varphi_2 \wedge \varphi_3) + \mathsf{Pr}(\varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3) + \mathsf{Pr}(\varphi_1 \wedge \neg\varphi_2 \wedge \varphi_3) + \mathsf{Pr}(\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3) = \\ &= \ldots = \sum_{\theta \in \Theta \text{ st } \theta \to \varphi_1} \mathsf{Pr}(\theta). \end{aligned} \tag{4.4}$$

It means that $\vdash_{(\Gamma,\Lambda)} \mathsf{Pr}(\varphi_1) = \sum_{\theta\in\Theta \text{ st } \theta\to\varphi_1} \mathsf{Pr}(\theta)$. We can obtain a similar formula for each $\varphi\in\Omega$. Moreover, since $\bigvee_{\theta\in\Theta}\theta \leftrightarrow \top$ and $\theta_i\wedge\theta_j\leftrightarrow\bot$ for any $\theta_i,\theta_j\in\Theta$, $\theta_i\neq\theta_j$, using axioms P2 and P4 we can deduce that $\mathsf{Pr}\left(\bigvee_{\theta\in\Theta}\theta\right) = \sum_{\theta\in\Theta}\mathsf{Pr}(\theta)$ and it follows that $\sum_{\theta\in\Theta}\mathsf{Pr}(\theta) = 1$. Before finishing, notice that PAux2 and P2 imply that

$$\vdash_{(\Gamma,\Lambda)} \bigwedge_{\theta\in\Theta}(\forall(\neg\theta)\to\mathsf{Pr}(\theta)=0).$$

Axiom P1 and the previous justifications, allow us to write (4.2) equivalently as:

$$\psi_j^1\wedge\cdots\wedge\psi_j^{n_j}\wedge\bigwedge_{\varphi\in\Omega}\left(\mathsf{Pr}(\varphi) = \sum_{\substack{\theta\in\Theta\\ \theta\to\varphi}}\mathsf{Pr}(\theta)\right)\wedge\left(\sum_{\theta\in\Theta}\mathsf{Pr}(\theta)=1\right)\wedge\left(\bigwedge_{\theta\in\Theta}\forall(\neg\theta)\to\mathsf{Pr}(\theta)=0\right)\wedge\bigwedge_{\theta\in\Theta}(\mathsf{Pr}(\theta)\geq 0). \tag{4.5}$$

Since we can assign probabilities independently to the different elements in Θ, (4.5) is satisfiable if and only if the system of inequalities (4.3) is satisfiable. Under the hypothesis that the system of inequalities is unsatisfiable, using the results of soundness and completeness for the axioms of inequality, the system would be inconsistent. But it would mean that we could derive an inconsistency from (4.5) using I1-I6, C1-C4, which is a contradiction with the consistency of (4.2). We conclude that the system (4.3) is satisfiable. Let $\{x_\theta^*\}_{\theta\in\Theta}$ be a solution.

The solution of (4.3) is used to define a probability distribution over the atoms and thus over the outcomes satisfying them. The probability distribution $\mathcal{P}:\mathsf{S}\to[0,1]$ is defined as follows:

$$\begin{cases} \mathcal{P}(\rho_\theta) &= x_\theta^*, \quad \text{for each } \theta\in\Theta,\\ \mathcal{P}(\rho) &= 0, \quad\;\; \text{for each } \rho\in\mathsf{S}\setminus\{\rho_\theta \mid \theta\in\Theta\}. \end{cases}$$

A probability space $\mathbb{P} = (\mathsf{S},\mathscr{A},\mu)$ is built on top of this probability distribution, considering the σ-algebra $\mathscr{A}$ generated by the set $\{\mathsf{S}^\varphi \mid \varphi\in\mathsf{Loc}\}$ and the probability measure

$$\begin{aligned} \mu:\mathscr{A} &\to [0,1]\\ X &\mapsto \sum_{\rho\in X}\mathcal{P}(\rho). \end{aligned}$$

Let us verify that μ is effectively a probability measure:

- Given $X\in\mathscr{A}$, $\mu(X)\geq 0$ since $\mu(X) = \sum_{\rho\in X}\mathcal{P}(\rho)$, and the system of inequalities (4.3) together with the definition of $\mathcal{P}$ implies that $\mathcal{P}(\rho)\geq 0$ for each $\rho\in\mathsf{S}$.

- We conclude that $\mu(\mathsf{S}) = 1$ by observing that $\mathsf{S}\in\mathscr{A}$ as result of $\mathsf{S} = \mathsf{S}^{t\approx t}$, and further $\mu(\mathsf{S}) = \sum_{\rho\in\mathsf{S}}\mathcal{P}(\rho)$, which leads to the expected measure 1 for the entire set of possible outcomes by simply recalling the definition of $\mathcal{P}$ and writing

$$\mu(\mathsf{S}) = \sum_{\rho\in\mathsf{S}}\mathcal{P}(\rho) = \sum_{\rho\in\mathsf{S}\setminus\{\rho^\theta\mid\theta\in\Theta\}}\mathcal{P}(\rho) + \sum_{\theta\in\Theta}\mathcal{P}(\rho_\theta) = 0 + \sum_{\theta\in\Theta}x_\theta^*.$$

Since $\{x^*_\theta\}_{\theta\in\Theta}$ is a solution for (4.3) we actually have $\mu(\mathsf{S}) = \sum_{\theta\in\Theta} x^*_\theta = 1$.

- Given a countable collection of pairwise disjoint sets $\{X_i\}_{i\in I} \subseteq \mathscr{A}$, the equality $\mu\left(\bigcup\limits_{i\in I} X_i\right) = \sum\limits_{i\in I} \mu(X_i)$ holds as a consequence of sets $\{X_i\}_{i\in I}$ being pairwise disjoint and from the following straightforward equalities,

$$\sum_{i\in I} \mu(X_i) = \sum_{i\in I}\sum_{\rho\in X_i} \mathcal{P}(\rho) = \sum_{\rho\in\bigcup\limits_{i\in I} X_i} \mathcal{P}(\rho) = \mu\left(\bigcup_{i\in I} X_i\right).$$

  Just note that each of the previous sums has a finite number of non-zero elements.

Now that an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ has emerged, it remains to prove that it actually satisfies all the relevant formulas in RelF. For that purpose, we leave an auxiliary remark whose proof follows easily by induction on the structure of φ.

Remark 4.3.3. Given $\neg\forall\varphi_0 \in \Xi$ and a local formula $\varphi \in \mathsf{Loc}$ with $\mathsf{names}(\varphi) = \tilde{n}$,

$$\forall[\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}} \in \Xi \text{ if and only if } \mathbb{A}, \mathrm{I}^{\mathbb{A}} \Vdash [\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}}.$$

We conclude the proof verifying that we have indeed a model for RelF. Recall that $\mathsf{RelF} \subseteq \forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc} \cup \mathsf{Prob} \cup \neg\mathsf{Prob}$, consider $\gamma \in \mathsf{RelF}$ and let us analyze the four cases:

- if γ is of the form $\forall\varphi$ with $\mathsf{names}(\varphi) = \tilde{n}$, we want to prove that for every $\rho \in \mathsf{S}$, $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \varphi$. Given $\rho \in \mathsf{S}$, recall that it was motivated by some $\neg\forall\varphi_0 \in \Xi$, say that $\rho = \rho^{\neg\forall\varphi_0}$. Since $\forall\varphi \in \mathsf{RelF} \subseteq \Xi$ it follows that $\forall[\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}} \in \Xi$ by construction of W. Using Remark 4.3.3 we conclude that $\mathbb{A}, \mathrm{I}^{\mathbb{A}} \Vdash [\varphi]^{\tilde{n}}_{\tilde{c}_{\varphi_0}}$, which according to definition of $\rho^{\neg\forall\varphi_0}$ implies that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^{\neg\forall\varphi_0} \Vdash_{\mathsf{loc}} \varphi$.

- if γ is of the form $\neg\forall\varphi$, with $\mathsf{names}(\neg\varphi) = \mathsf{names}(\varphi) = \tilde{n}$, notice that $\rho^{\neg\forall\varphi} \in \mathsf{S}$. Moreover, since $\neg\forall\varphi \in \Xi$, it follows that $\forall[\neg\varphi]^{\tilde{n}}_{\tilde{c}_\varphi} \in \Xi$. Remark 4.3.3 implies that $\mathbb{A}, \mathrm{I}^{\mathbb{A}} \Vdash [\neg\varphi]^{\tilde{n}}_{\tilde{c}_\varphi}$, which by definition of $\rho^{\neg\forall\varphi}$ leads to $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^{\neg\forall\varphi} \Vdash_{\mathsf{loc}} \neg\varphi$, so $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \neg\forall\varphi$.

- If $\gamma \in \mathsf{Prob}$ is of the form $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q$, we have:

$$\begin{array}{ll} & (\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \\ \text{iff} & q_1 \cdot \mu(\mathsf{S}^{\varphi_1}) + \cdots + q_\ell \cdot \mu(\mathsf{S}^{\varphi_\ell}) \geq q \\ \text{iff} & q_1 \sum\limits_{\rho\in\mathsf{S}^{\varphi_1}} \mathcal{P}(\rho) + \cdots + q_\ell \sum\limits_{\rho\in\mathsf{S}^{\varphi_\ell}} \mathcal{P}(\rho) \geq q \\ \text{iff} & q_1 \sum\limits_{\substack{\theta\in\Theta \text{ s.t.} \\ \theta\to\varphi_1}} \mathcal{P}(\rho_\theta) + \cdots + q_\ell \sum\limits_{\substack{\theta\in\Theta \text{ s.t.} \\ \theta\to\varphi_\ell}} \mathcal{P}(\rho_\theta) \geq q \\ \text{iff} & q_1 \sum\limits_{\substack{\theta\in\Theta \text{ s.t.} \\ \theta\to\varphi_1}} x^*_\theta + \cdots + q_\ell \sum\limits_{\substack{\theta\in\Theta \text{ s.t.} \\ \theta\to\varphi_\ell}} x^*_\theta \geq q. \end{array}$$

  The last inequality is valid since $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \in \mathsf{RelF}$ and $\{x^*_\theta\}_{\theta\in\Theta}$ is a solution for (4.3), hence the first assertion holds as well.

- If $\gamma \in \neg\mathsf{Prob}$ is of the form $\neg(q_1 \cdot \mathsf{Pr}(\varphi_1) + \dots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q)$, notice that

$$\begin{aligned}
& (\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \neg(q_1 \cdot \mathsf{Pr}(\varphi_1) + \dots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q) \\
\text{iff} \quad & (\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \not\Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \dots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q \\
\text{iff} \quad & q_1 \cdot \mu(\mathsf{S}^{\varphi_1}) + \dots + q_\ell \cdot \mu(\mathsf{S}^{\varphi_\ell}) < q \\
\text{iff} \quad & q_1 \sum_{\substack{\rho \in \mathsf{S} \text{ s.t.} \\ (\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \varphi_1}} \mathcal{P}(\rho) + \dots + q_\ell \sum_{\substack{\rho \in \mathsf{S} \text{ s.t.} \\ (\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \varphi_\ell}} \mathcal{P}(\rho) < q \\
\text{iff} \quad & q_1 \sum_{\substack{\theta \in \Theta \text{ s.t.} \\ \theta \to \varphi_1}} x^*_\theta + \dots + q_\ell \sum_{\substack{\theta \in \Theta \text{ s.t.} \\ \theta \to \varphi_\ell}} x^*_\theta < q
\end{aligned}$$

  Provided that $\{x^*_\theta\}_{\theta \in \Theta}$ is a solution for (4.3) and $\neg(q_1\mathsf{Pr}(\varphi_1) + \ldots + q_\ell\mathsf{Pr}(\varphi_\ell) \geq q \in \mathsf{RelF}$, the last expression holds.

We end the proof of completeness observing that all of this leads to our original motivation for the proof: $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \neg\delta$. □

This concludes the soundness and completeness results for DEqPrL. We are now able to use indistinctively the syntactic consequence relation $\vdash_{(\Gamma,\Lambda)}$ and the semantic consequence relation $\vDash_{\mathcal{I}_{(\Gamma,\Lambda)}}$ provided that the class $\mathcal{I}_{(\Gamma,\Lambda)}$ of intended interpretations is such that its algebras are axiomatized by a set Γ of Horn clauses over X and the corresponding interpretations for domain names are axiomatized by a finite set Λ of domain clauses of algebraic terms.

4.4 Decidability and Complexity

As observed for EqCL, DEqPrL cannot be expected to be decidable, as equational theories can easily be undecidable [13]. We show, however, that DEqPrL is decidable if we require the base equational theory to be convergent, and additionally the underlying domain clauses to have the subterm property. We further provide an automated procedure for decidability.

Again, we may wonder whether the logic would also be decidable with more general restrictions on the equational theory and on domain restrictions. It would be interesting to explore how general could be the underlying (decidable) equational theory in order to preserve the decidability of the logic. However, as the majority of the equational theories underlying information security examples are generated by convergent rewriting systems, we only focus on equational theories generated by convergent rewriting systems and in domain restrictions with the subterm property. All the more, we take advantage of the rewriting systems underlying equational theories to draw a decidability result. With this purpose, let us assume, from now on, that Γ is a convergent equational theory and Λ is a set of domain clauses with the subterm property.

4.4.1 Satisfiability

We devote this subsection to the analysis of the satisfiability problem for DEqPrL (SAT-DEqPrL). The SAT-DEqPrL problem consists in deciding the existence of a model for a global formula.

As we observed in the context of EqCL, satisfiability solvers usually require a particular format for the input formula. In this sense, we start by analyzing two satisfiability problems: the DNFSAT-DEqPrL is the satisfiability problem for DEqPrL whose input formula is in disjunctive normal form, whereas the CNFSAT-DEqPrL is the satisfiability problem for DEqPrL for which the input formula is required to be in conjunctive normal form. Afterwards, we proceed with the analysis of SAT-DEqPrL for arbitrary global formulas.

As a preliminary approach, easier to understand, we start by providing an algorithm that translates DNFSAT-DEqPrL problems into GenPSAT problems (introduced in Chapter 3). Afterwards, we extend the approach and provide a reduction of CNFSAT-DEqPrL to Satisfiability Modulo Theories (SMT). We end up this series of satisfiability results using a Tseitin-like transformation to analyse SAT-DEqPrL.

Moving to the propositional context

To describe an algorithm that reduces SAT-DEqPrL either to GenPSAT, to SAT or to SMT, we need to translate local formulas to the propositional context. We follow the same reasoning as for EqCL (see Section 2.4.12), but besides including equations into the propositional context, we should also incorporate the algebraic reasoning underlying the domain restrictions. In this sense, let us consider a set of propositional symbols corresponding to equations between nominal terms $\mathsf{Eq}(N)^{\mathsf{p}} = \{\mathsf{p}_{t_1 \approx t_2} \mid t_1, t_2 \in T(N)\}$ and a set of propositional symbols corresponding to domain restrictions $\mathsf{DRes}(N)^{\mathsf{p}} = \{\mathsf{p}_{t \in \mathrm{D}} \mid t \in T(N), \mathrm{D} \in \mathcal{D}\}$, and then define the translation of an arbitrary local formula $\varphi \in \mathsf{Loc}$ to a propositional formula prop_φ inductively, by:

- if φ is of the form $t_1 \approx t_2$, prop_φ is precisely $\mathsf{p}_{t_1 \approx t_2}$;
- if φ is of the form $t \in \mathrm{D}$ then prop_φ is $\mathsf{p}_{t \in \mathrm{D}}$;
- if φ is of the form $\neg\varphi_1$ then prop_φ is $\neg\mathsf{prop}_{\varphi_1}$;
- if φ is of the form $\varphi_1 \wedge \varphi_2$ then prop_φ is $\mathsf{prop}_{\varphi_1} \wedge \mathsf{prop}_{\varphi_2}$.

We also extend this propositional notation to probabilistic formulas: given a probabilistic formula δ of the form $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \bowtie q$ with $\bowtie \in \{\le, \ge, <, >\}$, prop_δ represents the probabilistic propositional formula $q_1 \cdot \mathsf{Pr}(\mathsf{prop}_{\varphi_1}) + \cdots + q_\ell \cdot \mathsf{Pr}(\mathsf{prop}_{\varphi_\ell}) \bowtie q$.

Once more, we must import the algebraic requirements underlying the equational reasoning in the presence of domain restrictions to the propositional context. For this purpose, assume that we want to test the satisfiability of $\delta \in \mathsf{Glob}$ and consider the set of relevant nominal terms for δ,

$$\mathsf{RelTerm}^{\delta} = \mathsf{subterms}(\{\delta\} \cup \Delta_{\in}^{\approx}) \cup \{t{\downarrow} \mid t \in \mathsf{subterms}(\{\delta\} \cup \Delta_{\in}^{\approx})\}\,,\ \text{where}$$

$\Delta_{\in}^{\approx} = \{\sigma(t) \approx \sigma(t') \mid (t \to t') \in R, \sigma \in \mathsf{subterms}(\delta)^{X}\} \cup \{\sigma(t) \textcircled{\in} \mathrm{D} \mid (t \textcircled{\in} \mathrm{D}) \in \mathrm{RHS}, \sigma \in \mathsf{subterms}(\delta)^{X}\}$ and $\mathrm{RHS} = \{t \textcircled{\in} \mathrm{D}'_1 \mid (t_1 \in \mathrm{D}_1, ..., t_{k_1} \in \mathrm{D}_{k_1} \Rightarrow t \textcircled{\in} \mathrm{D}'_1 ..., t \textcircled{\in} \mathrm{D}'_{k_2}) \in \Lambda\}$. $\mathsf{RelTerm}^{\delta}$ incorporates all the subterms of δ, their normal forms with respect to the convergent rewriting system R underlying Γ and, finally, the equational theory and the domain clauses instantiated on the subterms.

We achieve a sufficiently broad scope by defining the propositional symbols of interest as those that represent either equations between terms in $\mathsf{RelTerm}^{\delta}$ or domain restrictions for such terms, which are gathered in the set

$$\mathcal{B}^{\delta} = \mathcal{B}^{\mathsf{Eq}} \cup \mathcal{B}^{\mathsf{DRes}}, \tag{4.6}$$

where $\mathcal{B}^{\mathsf{Eq}} = \left\{\mathsf{p}_{t_1 \approx t_2} \mid t_1, t_2 \in \mathsf{RelTerm}^{\delta}\right\}$ and $\mathcal{B}^{\mathsf{DRes}} = \left\{\mathsf{p}_{t \in \mathrm{D}} \mid t \in \mathsf{RelTerm}^{\delta}, \mathrm{D} \in \mathcal{D}\right\}$. Both equational statements and domain restrictions must obey some relations to be imposed on their representatives. These restrictions are established in Φ^{δ}, defined as follows:

$$\begin{aligned}
\Phi^{\delta} = {} & \{\mathsf{p}_{t \approx t{\downarrow}} \mid t \in \mathsf{RelTerm}^{\delta}\} \cup \{\mathsf{p}_{t_1 \approx t_2} \to \mathsf{p}_{t_2 \approx t_1} \mid t_1, t_2 \in \mathsf{RelTerm}^{\delta}\} \cup \\
& \{(\mathsf{p}_{t_1 \approx t_2} \wedge \mathsf{p}_{t_2 \approx t_3}) \to \mathsf{p}_{t_1 \approx t_3} \mid t_1, t_2, t_3 \in \mathsf{RelTerm}^{\delta}\} \cup \\
& \{(\mathsf{p}_{t_1 \approx t'_1} \wedge ... \wedge \mathsf{p}_{t_n \approx t'_n}) \to \mathsf{p}_{f(t_1,...,t_n){\downarrow} \approx f(t'_1,...,t'_n){\downarrow}} \mid t_1, t'_1, ..., t_n, t'_n, f(t_1, ..., t_n)\downarrow, f(t'_1, ..., t'_n)\downarrow \in \mathsf{RelTerm}^{\delta}\} \cup \\
& \{(\mathsf{p}_{t_1 \approx t_2} \wedge \mathsf{p}_{t_1 \in \mathrm{D}}) \to \mathsf{p}_{t_2 \in \mathrm{D}} \mid t_1, t_2 \in \mathsf{RelTerm}^{\delta}, \mathrm{D} \in \mathcal{D}\} \cup \\
& \{\mathsf{p}_{\sigma(t_1){\downarrow} \in \mathrm{D}_1} \wedge ... \wedge \mathsf{p}_{\sigma(t_{k_1}){\downarrow} \in \mathrm{D}_{k_1}} \to \mathsf{p}_{\sigma(t){\downarrow} \textcircled{\in} \mathrm{D}'_1} \vee ... \vee \mathsf{p}_{\sigma(t){\downarrow} \textcircled{\in} \mathrm{D}'_{k_2}} \mid \sigma \in (\mathsf{RelTerm}^{\delta})^{X}, \\
& \qquad (t_1 \in \mathrm{D}_1, ..., t_{k_1} \in \mathrm{D}_{k_1} \Rightarrow t \textcircled{\in} \mathrm{D}'_1, ..., t \textcircled{\in} \mathrm{D}'_{k_2}) \in \Lambda\}.
\end{aligned} \tag{4.7}$$

We should emphasize that, since $\mathsf{subterms}(\delta)$ has linear size on the length of δ and the equational theory is convergent, $\mathsf{RelTerm}^{\delta}$ is well defined and has polynomial size on the length of δ. Denoting $\left|\mathsf{RelTerm}^{\delta}\right| = k$ and $|\mathcal{D}| = z$, Φ^{δ} has at most $k + k^2 + k^3 + k^{2a+2} + k^2 \cdot z + \lambda(k^{tmax} \cdot z^{dmax})$ elements, where a is the maximum arity of the function symbols occurring in $\mathsf{RelTerm}^{\delta}$, $|\Lambda| = \lambda$ and $tmax, dmax$ are the maximum number of terms and the maximum number of domain names occurring in a constraint in Λ. Sometimes we drop the superscript δ, provided that it is clear from context.

The subterm property provides control over the set Φ^{δ}, as long as it ensures that the domain restrictions over a term in $\mathsf{RelTerm}$ is only conditioned by domain restrictions over terms certainly in $\mathsf{RelTerm}$. Thus, elements in Φ^{δ} are the necessary to reason about the domain restrictions that influence δ.

DNFSAT-DEqPrL problem

The DNFSAT-DEqPrL problem consists in deciding the existence of a model for a global formula $\delta \in \mathsf{Glob}$ given in disjunctive normal form.

To test the satisfiability of a global formula $\delta \in \mathsf{Glob}$ given in DNF by $\bigvee_{j=1}^{m} \bigwedge_{i=1}^{n_j} \delta_i^j$, one computes the set $\mathcal{B}^\delta$ of propositional symbols described in (4.6), the set Φ^δ of propositional formulas described in (4.7) and then use Algorithm 4.1 to decide whether the given formula is satisfiable or not. Note that we should apply a satisfiability procedure to test the satisfiability of each disjunct. Each disjunct is a conjunction of global formulas either from $\forall\mathsf{Loc}$, $\neg\forall\mathsf{Loc}$ or $(\mathsf{Prob} \cup \neg\mathsf{Prob})$. Specifying explicitly those components, δ is given by:

$$\bigvee_{j=1}^{m} \left(\forall\varphi_1^j \wedge \ldots \wedge \forall\varphi_{n_j}^j \wedge \neg\forall\varphi_1^j \wedge \ldots \wedge \neg\forall\varphi_{k_j}^j \wedge \xi_1^j \wedge \ldots \wedge \xi_{s_j}^j \right).$$

Algorithm 4.1 DNFSAT-DEqPrL solver based on SAT and GenPSAT

```
procedure DNFSATDEqPrL
    input: DNF global formula $\delta$: $\bigvee_{j=1}^{m} \left( \forall\psi_1^j \wedge \ldots \wedge \forall\psi_{n_j}^j \wedge \neg\forall\varphi_1^j \wedge \ldots \wedge \neg\forall\varphi_{k_j}^j \wedge \xi_1^j \wedge \ldots \wedge \xi_{s_j}^j \right)$
    output: Sat or Unsat depending on whether $\delta$ is satisfiable or not
    for $j = 1$ to $m$ do                                        ▷ test each disjunct
        $H_j := \Phi^\delta \cup \{\mathsf{prop}_{\psi_1^j}, \ldots, \mathsf{prop}_{\psi_{n_j}^j}\}$                ▷ hard constraints
        $S_j := \{\mathsf{prop}_{\xi_1^j}, \ldots, \mathsf{prop}_{\xi_{s_j}^j}\}$                        ▷ soft constraints
        for $\ell = 1$ to $k_j$ do                         ▷ incorporate each $\neg\forall\varphi_\ell^j$ in $H_j$
            if sat_solver( $\bigwedge_{\mathsf{prop}_\psi \in H_j} \mathsf{prop}_\psi \wedge \neg\mathsf{prop}_{\varphi_\ell^j}$ ) == Unsat then
                break
        if $\ell == k_j + 1$ and genpsat_solver($H_j, S_j$) == Sat then
            return Sat    ▷ return Sat if all the iterations are successful for some disjunct
    return Unsat                    ▷ return Unsat if some iteration fails on every disjunct
```

Given a global formula $\delta \in \mathsf{Glob}$ in disjunctive normal form, the DNFSAT-DEqPrL tests the satisfiability of δ by reduction to both a GenPSAT solver and a SAT solver. The GenPSAT solver is represented in Algorithm 4.1 by an auxiliary procedure genpsat_solver and returns Sat or Unsat depending on wether the GenPSAT instance given as input is satisfiable or not; the SAT solver is represented by an auxiliary procedure sat_solver that returns Sat or Unsat depending on wether the input propositional formula is satisfiable or not. Notice that we have fixed a convergent equational theory Γ and a set of domain clauses Λ with the subterm property, so that the sets of propositional formulas $\mathcal{B}^\delta$ and Φ^δ are well defined and have polynomial length on the length of δ. Each disjunct is written as a conjunction of literals from $(\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc}) \cup (\mathsf{Prob} \cup \neg\mathsf{Prob})$. The qualitative literals (not involving probabilities) should incorporate the propositional set of the GenPSAT instance, whereas the quantitative literals constitute the probabilistic component of the GenPSAT instance. Recall that satisfying a formula of the form $\forall\varphi$ imposes that φ must be verified in all possible outcomes, whereas satisfying a formula like $\neg\forall\varphi$ requires that at least one possible outcome satisfies $\neg\varphi$. For this reason, the satisfiability of each disjunct is tested with a SAT solver in several iterations (one for each literal in $\neg\forall\mathsf{Loc}$) and then with a GenPSAT solver, to test the consistency of the probabilistic literals. When all iterations are successful for some disjunct, we conclude that δ is satisfiable.

Let us illustrate this simple algorithm with a simple example.

Example 4.4.1. Let us fix the signature $\mathrm{F}^{\mathrm{xor}}$, the equational theory Γ^{xor} and the axiomatization Λ^{xor} that we introduced in Example 4.1.1. We use Algorithm 4.1 to test the satisfiability of the DNF global formula:

$$(\mathsf{Pr}(n \approx \mathsf{zero}) \leq \tfrac{2}{3} \cdot \mathsf{Pr}(n \in \mathsf{even}) \wedge \forall n \in \mathsf{even} \wedge \neg\mathsf{Pr}(n \approx \mathsf{zero}) \leq \tfrac{2}{3}) \vee$$
$$\vee (\mathsf{Pr}(n \approx \mathsf{zero}) \leq \tfrac{2}{3} \cdot \mathsf{Pr}(n \in \mathsf{even}) \wedge \forall n \in \mathsf{even} \wedge \neg\forall \mathsf{suc}(n) \in \mathsf{odd}).$$

We start by observing that $\mathsf{RelTerm}^{\delta} = \{n, \mathsf{zero}, \mathsf{suc}(n)\}$ and stressing that, for instance, $(\mathsf{prop}_{n\in\mathsf{even}} \to \mathsf{prop}_{\mathsf{suc}(n)\in\mathsf{odd}}) \in \Phi^{\delta}$.

Testing the first disjunct of the formula, we come up with the GenPSAT instance (H_1, S_1) given by:

$$\begin{aligned} H_1 &= \Phi^{\delta} \cup \{\mathsf{prop}_{n\in\mathsf{even}}\}, \\ S_1 &= \{\mathsf{Pr}(\mathsf{prop}_{n\approx\mathsf{zero}}) \leq \tfrac{2}{3} \cdot \mathsf{Pr}(\mathsf{prop}_{n\in\mathsf{even}}), \mathsf{Pr}(\mathsf{prop}_{n\approx\mathsf{zero}}) > \tfrac{2}{3}\}, \end{aligned}$$

which is unsatisfiable provided that the classical propositional formulas are evaluated with probability 1, which derives an inconsistency. For the second iteration, we have:

$$\begin{aligned} H_2 &= \Phi^{\delta} \cup \{\mathsf{prop}_{n\in\mathsf{even}}\}, \\ S_2 &= \{\mathsf{Pr}(\mathsf{prop}_{n\approx\mathsf{zero}}) \leq \tfrac{2}{3} \cdot \mathsf{Pr}(\mathsf{prop}_{n\in\mathsf{even}})\}. \end{aligned}$$

Recalling that $(\mathsf{prop}_{n\in\mathsf{even}} \to \mathsf{prop}_{\mathsf{suc}(n)\in\mathsf{odd}}) \in \Phi^{\delta}$ and $\mathsf{prop}_{n\in\mathsf{even}} \in H_2$, we easily realize that this iteration is aborted provided that the classical propositional formula $\bigwedge_{\mathsf{prop}_{\psi}\in H_2} \mathsf{prop}_{\psi} \wedge \neg\mathsf{prop}_{\mathsf{suc}(n)\in\mathsf{odd}}$, that is sent to the SAT-solver, is not satisfiable.

Hence, we conclude that the given formula is unsatisfiable. △

The correctness of the satisfiability algorithm is addressed in the following Lemma.

Lemma 4.4.2. *If Γ is a convergent equational theory and Λ is a set of domain clauses with the subterm property, a global formula $\delta \in$ Glob in DNF is satisfiable iff Algorithm 4.1 returns* Sat.

Proving this Lemma requires showing that satisfiability at the propositional level carries over to DEqPrL. For this purpose, given $\delta \in \mathsf{Glob}$, we follow the lines of what was done in the proof of Lemma 2.4.3 and redefine the translation of outcomes with values in an F-algebra $\mathbb{A}$ with carrier set A to valuations in the propositional context. For this purpose, let $v_{(\cdot)}$ be the transformation of outcomes into valuations, $v_{(\cdot)} : A^N \to \{0,1\}^{\mathcal{B}}$ such that, given $\rho \in A^N$, the corresponding valuation $v_\rho : \mathcal{B} \to \{0,1\}$ is now defined by:

$$\begin{cases} v_\rho(\mathsf{p}_{t\in \mathrm{D}}) = 1 & \text{iff} \quad [\![t]\!]^\rho_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}) \\ v_\rho(\mathsf{p}_{t_1 \approx t_2}) = 1 & \text{iff} \quad [\![t_1]\!]^\rho_{\mathbb{A}} = [\![t_2]\!]^\rho_{\mathbb{A}}. \end{cases}$$

The proof that this translation is correct follows easily by induction on φ.

Lemma 4.4.3. *For each $\varphi \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and $\rho \in A^N$, $\mathbb{A}, \rho \Vdash_{\mathsf{loc}} \varphi$ iff $v_\rho\big(\mathsf{prop}_\varphi\big) = 1$.*

For the reciprocal implication, we need a small Lemma, that guarantees a kind of *finite model property* to the $\mathsf{GenPSAT}$ instances (H_j, S_j), with $j \in \{1, \dots, m\}$.

Lemma 4.4.4. *If there exists a probability distribution over $\{0,1\}^{\mathcal{B}^\delta}$ satisfying (H_j, S_j), then exists a probability distribution that also satisfies (H_j, S_j) but has a finite number of non-zero assignments of probabilities to valuations, for each $j \in \{1, \dots, m\}$.*

Proof. This proof is based on a reasoning very similar to what was done in the proof of completeness of DEqPrL. Let $j \in \{1, \dots, m\}$ and $\pi : \{0,1\}^{\mathcal{B}^\delta} \to [0,1]$ be a probability distribution that satisfies (H_j, S_j). We define a new probability distribution π' with a finite number of non-zero assignments over valuations by considering the j^{th} disjunct of δ:

$$\forall \psi_1^j \wedge \dots \wedge \forall \psi_{n_j}^j \wedge \neg \forall \varphi_1^j \wedge \dots \wedge \neg \forall \varphi_{k_j}^j \wedge \xi_1^j \wedge \dots \wedge \xi_{s_j}^j.$$

Let Ω be the set of the local formulas occurring inside the probabilistic literals $\xi_1^j, \dots, \xi_{s_j}^j$, $\Omega = \mathsf{lnPr}(\xi_1^j) \cup \dots \cup \mathsf{lnPr}(\xi_{s_j}^j)$, and define the suitable atoms

$$\Theta^{\mathsf{p}} = \left\{ \bigwedge_{\gamma \in \Upsilon} \mathsf{prop}_\gamma \wedge \bigwedge_{\omega \in \Omega \setminus \Upsilon} \neg \mathsf{prop}_\omega \mid \Upsilon \subseteq \Omega \cup \{\psi_1^j, \dots, \psi_{n_j}^j\} \right\}.$$

We assign a valuation v_θ for each $\theta \in \Theta^{\mathsf{p}}$ such that $V^\theta = \{v \in \{0,1\}^{\mathcal{B}^\delta} \mid v(\theta) = 1\} \neq \varnothing$ and consider the set of valuations $V \subseteq \{0,1\}^{\mathcal{B}^\delta}$ arising from this construction:

$$V = \{v_\theta \mid \theta \in \Theta \text{ and } V^\theta \neq \varnothing\}.$$

Notice that $V \neq \varnothing$ due to the suitable construction of Θ, which implies that $\bigcup_{\theta \in \Theta} V^\theta = \{0,1\}^{\mathcal{B}^\delta}$. A probability distribution $\pi' : \{0,1\}^{\mathcal{B}^\delta} \to [0,1]$ is then defined based on π,

$$\begin{cases} \pi'(v_\theta) & = & \sum\limits_{v \in V^\theta} \pi(v), & \text{for each } \theta \in \Theta \text{ such that } V^\theta \neq \varnothing \\ \pi'(v) & = & 0, & \text{for the remaining } v \in \{0,1\}^{\mathcal{B}^\delta} \setminus V. \end{cases}$$

It remains to prove that π' is actually a probability distribution over $\{0,1\}^{\mathcal{B}^\delta}$: obviously it is always non-negative, i.e., $\pi'(v) \geq 0$ for every valuation v, and only assigns a finite number of non-zero probabilities to valuations; then notice that $\bigcup\limits_{\theta\in\Theta} V^\theta = \{0,1\}^{\mathcal{B}^\delta}$ and $V^{\theta_1} \cap V^{\theta_2} = \varnothing$ whenever $\theta_1 \neq \theta_2$, and let us check that the probability distribution sums up to 1:

$$1 = \sum_{v\in\{0,1\}^{\mathcal{B}^\delta}} \pi(v) = \sum_{\theta\in\Theta} \pi'(v_\theta) = \sum_{v\in V} \pi'(v) + \sum_{v\in\{0,1\}^{\mathcal{B}^\delta}\setminus V} \pi'(v) = \sum_{v\in\{0,1\}^{\mathcal{B}^\delta}} \pi'(v).$$

To conclude that the probability distribution π' satisfies the $\mathsf{GenPSAT}$ instance (H_j, S_j) notice that, given $\psi \in \Omega \cup \{\psi_1^j, \dots, \psi_{n_j}^j\}$,

$$\sum_{v\in\{0,1\}^{\mathcal{B}^\delta}} v(\psi)\cdot\pi'(v) = \sum_{\substack{v_\theta\in V\\ \theta\to\psi}} \pi'(v_\theta) = \sum_{v\in V} v(\psi)\cdot\pi(v).$$

Hence, π' satisfies (H_j, S_j). □

In order to proceed with the reciprocal implication, we will need to keep safe the propositional information while defining a suitable F-structure satisfying δ. For this purpose, let us consider an appropriate finite set of valuations $V \subseteq \{0,1\}^{\mathcal{B}^\delta}$ containing the valuations satisfying $\bigwedge\limits_{\mathsf{prop}_\psi\in H_j} \mathsf{prop}_\psi \wedge \neg\mathsf{prop}_{\varphi_\ell^j}$, for each $\ell \in \{1,\dots,k_j\}$ and some others for which are assigned non-zero probabilities that constitute a probability distribution satisfying the $\mathsf{GenPSAT}$ instance (H_j, S_j).

We outline a candidate F-algebra for a model for δ by extending the signature F with new constants $c_{v,n}$, for each $n \in N$ and $v \in V$, that should keep hold the propositional behaviour into the logic: the extended signature F^* coincides with F in all but

$$\mathrm{F}_0^* = \mathrm{F}_0 \cup \bigcup_{v\in V} \{c_{v,n} \mid n \in N\}. \tag{4.8}$$

In order to keep safe the propositional information, we collect a record, into the logic, of all propositional symbols provided each valuation. For the purpose, we consider an enumeration of the set $\mathcal{B}\times V$ indexed by $i \in I \subseteq \mathbb{N}$, and save such information in the set $M = \bigcup\limits_{i\in I} M_i$, defined inductively by:

$$\begin{aligned} M_0 = \quad & \{\forall(\sigma(t) \approx \sigma(t')) \mid t \approx t' \in \Gamma, \sigma \in \mathbb{T}_{\mathrm{F}^*}(\varnothing)^X\} \cup \\ & \{\forall(\sigma(t_1) \in \mathrm{D}_1 \wedge \dots \wedge \sigma(t_{k_1}) \in \mathrm{D}_{k_1} \to \sigma(t) \circledin \mathrm{D}_1' \vee \dots \vee \sigma(t) \circledin \mathrm{D}_{k_2}') \mid \\ & \sigma \in \mathbb{T}_{\mathrm{F}^*}(\varnothing)^X, (t_1 \in \mathrm{D}_1, \dots, t_{k_1} \in \mathrm{D}_{k_1} \to t \circledin \mathrm{D}_1', \dots, t \circledin \mathrm{D}_{k_2}') \in \Lambda\} \end{aligned} \tag{4.9}$$

$$M_{i+1} = \begin{cases} M_i \;\cup\; \{\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}\}, & \text{if } v_i(\mathsf{p}_{\varphi_i}) = 1 \\ M_i \;\cup\; \{\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}\}, & \text{if } v_i(\mathsf{p}_{\varphi_i}) = 0 \end{cases} \quad \text{for } i \in I \text{ and } \tilde{n} = names(\varphi_i).$$

Remark 4.4.5. Notice that by $\mathsf{N2}$, $\mathsf{N3}$: $\vdash_{(\Gamma,\Lambda)} \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n} \leftrightarrow \neg\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}$.

We now address the consistency of $M = \bigcup_{i \in I} M_i \subseteq \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$ regarding the extended signature F^* and an empty set of names.

Lemma 4.4.6. *If Algorithm 4.1 returns* Sat*, then* $M = \bigcup\limits_{i \in I} M_i \subseteq \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$ *is consistent.*

Remark 4.4.7. Given $t_1, t_2 \in \mathsf{RelTerm}^{\delta}$, we have $\mathsf{p}_{t_1 \approx t_2} \in \mathcal{B}^{\delta}$. Hence, considering a valuation $v \in V$ we necessarily have that either $\forall([t_1]^{\widetilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \approx [t_2]^{\widetilde{n_{12}}}_{\tilde{c}_{v,n_{12}}}) \in M$ or $\forall([t_1]^{\widetilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \not\approx [t_2]^{\widetilde{n_{12}}}_{\tilde{c}_{v,n_{12}}}) \in M$, where $\widetilde{n_{12}} = \mathsf{names}(t_1) \cup \mathsf{names}(t_2)$. The same holds for the domain restrictions: given $\mathrm{D} \in \mathcal{D}$, either $\forall([t_1]^{\tilde{n}}_{\tilde{c}_{v,n}} \in \mathrm{D}) \in M$ or $\forall([t_1]^{\tilde{n}}_{\tilde{c}_{v,n}} \notin \mathrm{D}) \in M$, where $\tilde{n} = \mathsf{names}(t_1)$.

We prepare the proof of Lemma 4.4.6 by considering a substitution of constants back to names,

$$\begin{array}{rrcl} \varrho : & \{c_{v,n} \mid n \in \mathbb{N}, v \in V\} & \longrightarrow & N \\ & c_{v,n} & \longmapsto & n \ . \end{array}$$

Let us abuse notation and denote by ϱ the extension of this substitution to the set of local formulas over the extended signature F^* and an empty set of names, $\mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$.

Lemma 4.4.8. *Let* M_0 *be the set of global formulas defined in* (4.9)*. Given a local formula* $\varphi \in \mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$ *such that* $\mathsf{subterms}(\varrho(\varphi)) \subseteq \mathsf{RelTerm}^{\delta}$*, we have:*

$$M_0 \vdash_{(\Gamma,\Lambda)} \forall\varphi \text{ iff } \Phi^{\delta} \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}.$$

Proof. The reciprocal implication is immediate by recalling the definition of Φ^{δ} and based on axioms from $\mathcal{H}_{(\Gamma,\Lambda)}$, hence we focus on the direct implication.

Assume that $M_0 \vdash_{(\Gamma,\Lambda)} \forall\varphi$ and let us prove, by induction on the structure of this deduction, that each case leads to $\Phi^{\delta} \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}$.

- If $\forall\varphi \in M_0$ then φ is either of the form

  $$\sigma(t) \approx \sigma(t') \quad \text{or} \quad \sigma(t_1) \in D_1 \wedge \ldots \wedge \sigma(t_{k_1}) \in D_{k_1} \to \sigma(t) \textcircled{$\in$} D'_1 \vee \ldots \vee \sigma(t) \textcircled{$\in$} D'_{k_2}$$

  for some $\sigma \in T_{\mathrm{F}^*}(\varnothing)^X$, but then $\varrho \circ \sigma \in T(N)^X$ and the following cases hold, respectively:

  (i) either $(t \approx t') \in \Gamma$, and $(\mathsf{p}_{\varrho\circ\sigma(t) \approx \varrho\circ\sigma(t)\downarrow}), (\mathsf{p}_{\varrho\circ\sigma(t') \approx \varrho\circ\sigma(t')\downarrow}) \in \Phi^{\delta}$, but since $\varrho \circ \sigma(t)\downarrow = \varrho \circ \sigma(t')\downarrow$ it follows that $\Phi^{\delta} \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{p}_{\varrho\circ\sigma(t) \approx \varrho\circ\sigma(t')}$,

  (ii) or $(t_1 \in D_1, \ldots t_{k_1} \in D_{k_1} \to t \textcircled{$\in$} D'_1, \ldots, t \textcircled{$\in$} D'_{k_2}) \in \Lambda$ and so the propositional formula $(\mathsf{p}_{\varrho\circ\sigma(t_1)\downarrow} \in D_1 \wedge \ldots \wedge \mathsf{p}_{\varrho\circ\sigma(t_{k_1})\downarrow} \in D_{k_1} \to \mathsf{p}_{\varrho\circ\sigma(t)\downarrow \textcircled{$\in$} D'_1} \vee \ldots \vee \mathsf{p}_{\varrho\circ\sigma(t)\downarrow \textcircled{$\in$} D'_{k_2}}) \in \Phi^{\delta}$, which implies that $(\mathsf{p}_{\varrho\circ\sigma(t_1)\downarrow \in D_1} \wedge \ldots \wedge \mathsf{p}_{\varrho\circ\sigma(t_{k_1})\downarrow \in D_{k_1}} \to \mathsf{p}_{\varrho\circ\sigma(t)\downarrow \textcircled{$\in$} D'_1} \vee \ldots \vee \mathsf{p}_{\varrho\circ\sigma(t)\downarrow \textcircled{$\in$} D'_{k_2}}) \in \Phi^{\delta}$ that, together with the fact that $\mathsf{p}_{\varrho\circ\sigma(t') \approx \varrho\circ\sigma(t')\downarrow} \in \Phi^{\delta}$ and since for each $t' \in \{t_1 \ldots, t_{k_1}, t\}$, $\Phi^{\delta} \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{p}_{\varrho\circ\sigma(t') \approx \varrho\circ\sigma(t')\downarrow} \to (\mathsf{p}_{\varrho\circ\sigma(t') \in D} \leftrightarrow \mathsf{p}_{\varrho\circ\sigma(t')\downarrow \in D})$, allows us to conclude that:

  $$\Phi^{\delta} \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{p}_{\varrho\circ\sigma(t_1) \in D_1} \wedge \ldots \wedge \mathsf{p}_{\varrho\circ\sigma(t_{k_1}) \in D_{k_1}} \to \mathsf{p}_{\varrho\circ\sigma(t) \textcircled{$\in$} D'_1} \vee \ldots \vee \mathsf{p}_{\varrho\circ\sigma(t) \textcircled{$\in$} D'_{k_2}}.$$

- If $\forall\varphi$ is an axiom, then it is either an instance of:
  - ◇ $\mathsf{Eq1}$-$\mathsf{Eq4}$, where we immediately conclude that $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}$;
  - ◇ DEq which would imply $\mathsf{prop}_{\varrho(\varphi)} \in \Phi^\delta$;
  - ◇ $\mathsf{D}(\Lambda)$, which reduces to the analysis in (ii);
  - ◇ $\mathsf{E}(\Gamma)$, which reduces to the analysis in (i);
  - ◇ $\mathsf{EqC1}$-$\mathsf{EqC4}$, where we conclude that $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}$ as an immediate application of the axioms of propositional logic.

- If $\forall\varphi$ results from applying inference rule $\mathsf{C4}$ to δ_1 and $\delta_1 \to \delta_2$, then δ_2 should coincide with $\forall\varphi$ and $M_0 \vdash_{(\Gamma,\Lambda)} \{\delta_1, \delta_1 \to \delta_2\}$. We have several cases to analyze, we overview some of them:
  - ◇ if $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq2}$-$\mathsf{Eq4}$ (after the application of N and $\mathsf{C4}$), then the deduction is clearly deducible in the propositional context since Φ^δ contains all such properties;
  - ◇ the case where $\forall\varphi$ is deduced using $\mathsf{EqC1}$-$\mathsf{EqC4}$ (after applying N and $\mathsf{C4}$), or $\mathsf{C1}$-$\mathsf{C3}$, is covered by the propositional reasoning, leading to $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}$;
  - ◇ deductions from $\mathsf{D}(\Lambda)$ are under the scope of the propositional context as we saw before;
  - ◇ the propositional analogue of DEq in Φ^δ allows us to conclude that $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi)}$ when $\delta_1 \to \delta_2$ is an instance of DEq (after applying N and $\mathsf{C4}$);
  - ◇ if $\delta_1 \to \delta_2$ follows from $\mathsf{N1}$ and takes the form $\forall\varphi_1 \wedge \forall\varphi_2 \to \forall(\varphi_1 \wedge \varphi_2)$, since $\varphi_1 \wedge \varphi_2 \in \mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$, then $\varphi_1, \varphi_2 \in \mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$, and from $M_0 \vdash_{(\Gamma,\Lambda)} \forall\varphi_1 \wedge \forall\varphi_2$, by induction hypothesis, we have $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi_1)} \wedge \mathsf{prop}_{\varrho(\varphi_2)}$, so $\Phi^\delta \vdash_{\mathcal{H}_{\mathsf{CPL}}} \mathsf{prop}_{\varrho(\varphi_1 \wedge \varphi_2)}$;
  - ◇ if $\delta_1 \to \delta_2$ follows from $\mathsf{N1}$ but is of the form $\forall(\varphi_1 \wedge \varphi_2) \to \forall\varphi_i$ for some $i \in \{1,2\}$, then it is straightforward;
  - ◇ if $\delta_1 \to \delta_2$ is an instance of $\mathsf{N3}$ then it is of the form $\neg\forall\psi \to \forall\neg\psi$ and since $\neg\psi \in \mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$, then $\psi \in \mathsf{Loc}^{\varnothing}_{\mathrm{F}^*}$; recall that $M_0 \vdash_{(\Gamma,\Lambda)} \neg\forall\psi$ so, by induction hypothesis, $\Phi^\delta \nvdash \mathsf{prop}_{\varrho(\psi)}$ which means that $\Phi^\delta \vdash \mathsf{prop}_{\varrho(\neg\psi)}$. □

Proof of Lemma 4.4.6. Assume that Algorithm 4.1 returns Sat and let us prove, by induction on i, that M_i is consistent for every $i \in I$. Let us begin observing that M_0 is consistent since otherwise we would derive $\perp$ from M_0 and, in that case, using Lemma 4.4.8, we would be able to deduce $\mathsf{prop}_\perp$ from Φ^δ, which is a contradiction with the initial assumption that Algorithm 4.1 returns Sat.

Let $i \in I$ and assume that M_{i-1} is consistent but M_i is inconsistent. We analyze both cases that potentially lead to the inconsistency: $v_i(\mathsf{p}_{\varphi_i}) = 1$, $v_i(\mathsf{p}_{\varphi_i}) = 0$.

Case 1: Assume $v_i(\mathsf{p}_{\varphi_i}) = 1$. Since M_i is inconsistent, $M_i \vdash_{(\Gamma,\Lambda)} \delta$ for any $\delta \in \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$. In particular:

$$M_i \vdash_{(\Gamma,\Lambda)} \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$$

which, since $M_i = M_{i-1} \cup \{\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}\}$ and recalling Remark 4.4.5, implies that $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$. Let us check that it would lead to $v_i(\mathsf{p}_{\varphi_i}) = 0$, which is a contradiction. Recalling that either $\varphi_i \in \mathsf{Eq}_{\mathrm{F}}(N)$ or $\varphi_i \in \mathsf{DRes}_{\mathrm{F}}(N)$, we can use induction on the structure of the proof to conclude the first case.

- If $\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \in M_{i-1}$ then it is a consequence of originally $\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \in M_0$: notice that for different valuations the constants are distinct, so $[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \neq [\psi]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$ for any $v \neq v_i$ and for any $\psi \in \mathsf{Loc}$; for different formulas $\psi \neq \varphi_i$ we should also have $[\psi]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \neq [\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$, since the constants are different for distinct names. But then, since $\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \in M_0$, $\mathsf{p}_{\varphi_i} \in \mathcal{B}^\delta$ and v_i satisfies Φ^δ we should have $v_i(\mathsf{p}_{\varphi_i}) = 0$.
- Since $\varphi_i \in \mathsf{Eq}(N) \cup \mathsf{DRes}(N)$, the only chance for $\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$ to be an instance of an axiom would be $\mathsf{D}(\Lambda)$, which would reduce us to the previous case.
- If it is the result of applying $\mathsf{C4}$ to δ_1, $\delta_1 \to \delta_2$, then δ_2 should coincide with $\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$ and $M_{i-1} \vdash_{(\Gamma,\Lambda)} \{\delta_1, \delta_1 \to \delta_2\}$. We have several cases depending on $\delta_1 \to \delta_2$. We analyze some of them and the others are left to the reader:
  - ◇ If $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq2}$ (together with N, $\mathsf{C3}$ and $\mathsf{C4}$) then φ_i should be of the form $t_1 \approx t_2$. Since $t_1, t_2 \in \mathsf{RelTerm}^\delta$, the propositional representation of $\mathsf{Eq2}$ is in Φ^δ, and v_i satisfies Φ^δ. Notice yet that δ_1 is of the form $\forall\neg[t_2 \approx t_1]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$ and by induction hypothesis $v_i(\mathsf{p}_{t_2 \approx t_1}) = 0$, so $v_i(\mathsf{p}_{t_1 \approx t_2}) = 0$. Analogously, we can address the case were $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq4}$.
  - ◇ If $\delta_1 \to \delta_2$ is an instance of classical axiom $\mathsf{EqC2}$ (applying N and $\mathsf{C3}$) then δ_1 should be of the form $\neg(\psi \to \forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}})$ which means $M_{i-1} \vdash_{(\Gamma,\Lambda)} \psi \wedge \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$ and it follows that $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$, which by induction hypothesis implies $v_i(\mathsf{p}_{\varphi_i}) = 0$.
  - ◇ If $\delta_1 \to \delta_2$ is an instance of $\mathsf{EqC4}$ (with N) then δ_1 should be of the form $\neg((\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \to \forall\varphi) \to \forall\varphi)$ which implies $M_{i-1} \vdash_{(\Gamma,\Lambda)} (\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}} \to \forall\varphi) \wedge \neg\forall\varphi$, that can be written as $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}}$. By induction hypothesis it follows that $v_i(\mathsf{p}_{\varphi_i}) = 0$.

Case 2: Assume $v_i(\mathsf{p}_{\varphi_i}) = 0$. Since M_i is inconsistent, $M_i \vdash_{(\Gamma,\Lambda)} \delta$ for any $\delta \in \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$. In particular:

$$M_i \vdash_{(\Gamma,\Lambda)} \forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i,n}},$$

which, since $M_i = M_{i-1} \cup \{\forall\neg[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}\}$ and recalling Remark 4.4.5, implies that $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}$. We check, again by induction on the structure of the proof, that it implies $v_i(\mathsf{p}_{\varphi_i}) = 1$, which is a contradiction. Note that $\varphi_i \in \mathsf{Eq}_{\mathrm{F}}(N)$ or $\varphi_i \in \mathsf{DRes}_{\mathrm{F}}(N)$.

- If $\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n} \in M_{i-1}$ then it should result from $\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n} \in M_0$ by the same reason as for case 1. And, the same way, we are able to conclude that $v_i(\mathsf{p}_{\varphi_i}) = 1$.
- Since $\varphi_i \in \mathsf{Eq}(N) \cup \mathsf{DRes}(N)$, $\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}$ could be an instance of $\mathsf{E}(\Gamma)$ or $\mathsf{D}(\Lambda)$ (which would report us to the previous case) or an instance of $\mathsf{Eq1}$. In the last case φ_i should be of the form $t \approx t$ for some $t \in \mathsf{RelTerm}^{\delta}$. Since v_i satisfies Φ^{δ}, we would have $v_i(\mathsf{p}_{\varphi_i}) = 1$.
- If it is the result of applying $\mathsf{C4}$ to δ_1, $\delta_1 \to \delta_2$ then δ_2 should coincide with $\forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}$ and $M_{i-1} \vdash_{(\Gamma,\Lambda)} \{\delta_1, \delta_1 \to \delta_2\}$. We have several cases depending on $\delta_1 \to \delta_2$. Again, we analyze some of them:
  - ◇ If $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq2}$ (applying N), then φ_i is of the form $t_2 \approx t_1$ with $t_1, t_2 \in \mathsf{RelTerm}^{\delta}$, and δ_1 should be of the form $\forall[t_1 \approx t_2]^{\tilde{n}}_{\tilde{c}_{v_i},n}$, which by induction hypothesis would imply $v_i(\mathsf{p}_{t_1 \approx t_2}) = 1$, and so $v_i(\mathsf{p}_{\varphi_i}) = 1$.
  - ◇ If $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq3}$ (applying N), then φ_i should be $t_1 \approx t_3$ and δ_1 should be $\forall(t_1 \approx t_2) \wedge \forall(t_2 \approx t_3)$. But notice that $t_1\downarrow, t_3\downarrow \in \mathsf{RelTerm}^{\delta}$ and $t_2\downarrow = t_1\downarrow = t_3\downarrow$, so we would have, by reflexivity and transitivity that $v_i(\mathsf{p}_{t_1 \approx t_2\downarrow} \wedge \mathsf{p}_{t_2\downarrow \approx t_3}) = 1$, which together with the propositional representative of transitivity in Φ^{δ} implies $v_i(\mathsf{p}_{t_1 \approx t_3}) = 1$.
  - ◇ If $\delta_1 \to \delta_2$ is an instance of $\mathsf{Eq4}$ (applying N), then φ_i is an equation of the form $f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$ with $t_1, \ldots, t_n, t'_1, \ldots, t'_n \in \mathsf{RelTerm}^{\delta}$ and δ_1 is given by $\forall(t_1 \approx t'_1) \wedge \ldots \wedge \forall(t_n \approx t'_n)$ (if some t_i or t'_j is a normal form, consider in its substitution the subterm of δ that originated it). By induction hypothesis $v_i(\mathsf{p}_{t_1 \approx t'_1} \wedge \ldots \wedge \mathsf{p}_{t_n \approx t'_n}) = 1$, which together with the propositional representative of $\mathsf{Eq4}$ implies that $v_i(\mathsf{p}_{\varphi_i}) = 1$.
  - ◇ If $\delta_1 \to \delta_2$ is a consequence of $\mathsf{N1}$, then δ_1 is of the form $\forall([\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n} \wedge \psi_2)$. But notice that $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall([\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n} \wedge \psi_2)$ implies $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall[\varphi_i]^{\tilde{n}}_{\tilde{c}_{v_i},n}$, which by induction hypothesis implies $v_i(\mathsf{p}_{\varphi_i}) = 1$.
  - ◇ If $\delta_1 \to \delta_2$ is an instance of DEq (applying N), then φ_i is of the form $t_2 \in D$ with $t_2 \in \mathsf{RelTerm}^{\delta}$ and δ_1 is $\forall(t_1 \approx t_2) \wedge \forall(t_1 \in D)$. Notice that $M_{i-1} \vdash_{(\Gamma,\Lambda)} \delta_1$, i.e., $M_{i-1} \vdash_{(\Gamma,\Lambda)} \forall(t_1 \approx t_2) \wedge \forall(t_1 \in D)$ and by induction hypothesis we have $v_i(\mathsf{p}_{t_1\downarrow \approx t_2} \wedge \mathsf{p}_{t_1\downarrow \in D}) = 1$, which by the propositional representative of Φ^{δ} implies that $v_i(\mathsf{p}_{t_2 \in D}) = 1$.

Both cases lead to a contradiction, so M_i should be consistent for each $i \in I$, and so M is consistent regarding F^*. $\square$

We are now ready to use all of these technical details in the proof of Lemma 4.4.2.

Proof of Lemma 4.4.2. Assume that Γ is a convergent equational theory, Λ is a set of domain clauses with the subterm property and let $\delta \in \mathsf{Glob}$ be any global formula in DNF. To prove the direct implication, consider an F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ with $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$ and satisfying δ: $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \bigvee_{j=1}^{m} \bigwedge_{i=1}^{n_j} \delta_i^j$. This means that there exists $j \in \{1, \dots, m\}$ such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \bigwedge_{i=1}^{n_j} \delta_i^j$. Since each δ_i^j is a literal, we can rewrite it as

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall\psi_1^j \wedge \dots \wedge \forall\psi_{n_j}^j \wedge \neg\forall\varphi_1^j \wedge \dots \wedge \neg\forall\varphi_{k_j}^j \wedge \xi_1^j \wedge \dots \wedge \xi_{s_j}^j. \tag{4.10}$$

We need to verify that for each $\ell \in \{1, \dots, k_j\}$, there exists a valuation satisfying $\bigwedge_{\mathsf{prop}_\psi \in H_j} \mathsf{prop}_\psi \wedge \neg\mathsf{prop}_{\varphi_\ell^j}$ and, furthermore, that there exists a probability distribution π over the set of all valuations $\{0,1\}^{\mathcal{B}^\delta}$, satisfying the $\mathsf{GenPSAT}$ instance (H_j, S_j).

For the first assertion, let $\rho^{\neg\forall\varphi_\ell^j}$ be the outcome such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^{\neg\forall\varphi_\ell^j} \Vdash_{\mathsf{loc}} \neg\varphi_\ell^j$. Notice that (4.10) implies that for each $s \in \{1, \dots, n_j\}$, $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^{\neg\forall\varphi_\ell^j} \Vdash_{\mathsf{loc}} \psi_s^j$. Using Lemma 4.4.3, it is immediate to conclude that $v_{\rho^{\neg\forall\varphi_\ell^j}}$ satisfies $\bigwedge_{\mathsf{prop}_\psi \in H_j} \mathsf{prop}_\psi \wedge \neg\mathsf{prop}_{\varphi_\ell^j}$.

For the latter assertion, notice that we can assume that there exists at least one probabilistic literal in (4.10). Otherwise, testing (H_j, S_j) would be equivalent to test H_j in SAT and we were done. That said, we need to find out a probability distribution over the set of all valuations $\{0,1\}^{\mathcal{B}^\delta}$ satisfying (H_j, S_j). Again, we follow a reasoning similar to the proof of completeness of DEqPrL: collect in Ω the local formulas occurring inside the probabilistic literals $\xi_1^j, \dots, \xi_{s_j}^j$, $\Omega = \mathsf{InPr}(\xi_1^j) \cup \dots \cup \mathsf{InPr}(\xi_{s_j}^j)$, define the suitable atoms $\Theta = \left\{ \bigwedge_{\gamma \in \Upsilon} \gamma \wedge \bigwedge_{\omega \in \Omega \setminus \Upsilon} \neg\omega \mid \Upsilon \subseteq \Omega \right\}$, and then assign an outcome ρ_θ for each $\theta \in \Theta$ such that $\mathsf{S}^\theta \neq \varnothing$. Now consider the set of valuations $V \subseteq \{0,1\}^{\mathcal{B}^\delta}$ arising from this construction:

$$V = \{v_{\rho_\theta} \mid \theta \in \Theta \text{ and } \mathsf{S}^\theta \neq \varnothing\}. \tag{4.11}$$

Notice that $V \neq \varnothing$ due to the suitable construction of Θ, which implies that $\bigcup_{\theta \in \Theta} \mathsf{S}^\theta = \mathsf{S} \neq \varnothing$.

A probability distribution $\pi : \{0,1\}^{\mathcal{B}^\delta} \to [0,1]$ is then defined based on μ,

$$\begin{cases} \pi(v_{\rho_\theta}) = \mu(\mathsf{S}^\theta), & \text{for each } \theta \in \Theta \text{ such that } \mathsf{S}^\theta \neq \varnothing \\ \pi(v) = 0, & \text{for the remaining } v \in \{0,1\}^{\mathcal{B}^\delta} \setminus V. \end{cases}$$

It remains to prove that π is actually a probability distribution over $\{0,1\}^{\mathcal{B}^\delta}$: obviously it is always non-negative, i.e., $\pi(v) \geq 0$ for every valuation v; then recall that $\bigcup_{\theta \in \Theta} \mathsf{S}^\theta = \mathsf{S}$ and

$\mathsf{S}^{\theta_1} \cap \mathsf{S}^{\theta_2} = \varnothing$ whenever $\theta_1 \neq \theta_2$, and let us check that the probability distribution sums up to 1:

$$1 = \sum_{\theta \in \Theta} \mu(\mathsf{S}^\theta) = \sum_{\theta \in \Theta} \pi(v^{\rho_\theta}) = \sum_{v \in V} \pi(v) + \sum_{v \in \{0,1\}^{\mathcal{B}^\delta} \setminus V} \pi(v) = \sum_{v \in \{0,1\}^{\mathcal{B}^\delta}} \pi(v).$$

Once defined the probability distribution, let us check that the $\mathsf{GenPSAT}$ instance (H_j, S_j) is satisfiable, i.e., each propositional formula in H_j is satisfied with probability 1 and the probability formulas in S_j are also satisfied. To ease notation, let us use W^φ to denote the set of valuations in $W \subseteq \{0,1\}^{\mathcal{B}^\delta}$ satisfying prop_φ, $W^\varphi = \{v \in W \mid v(\mathsf{prop}_\varphi) = 1\}$.

- For each $s \in \{1, \ldots, n_j\}$, we have $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall \psi_s^j$, hence we also have $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \mathsf{Pr}(\psi_s^j) = 1$. The successive application of $\mathsf{P2}$, as was remarked in (4.4), allows us to write it equivalently as $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \sum_{\theta \in \Theta \text{ s.t. } \theta \to \psi_s^j} \mathsf{Pr}(\theta) = 1$, i.e., $\sum_{\theta \in \Theta \text{ s.t. } \theta \to \psi_s^j} \mu(\mathsf{S}^\theta) = 1$. But then,

$$\sum_{v \in \{0,1\}^{\mathcal{B}^\delta}} v\left(\mathsf{prop}_{\psi_s^j}\right) \cdot \pi(v) = \sum_{v \in V^{\psi_s^j}} \pi(v) = \sum_{\theta \in \Theta \text{ s.t. } \theta \to \psi_s^j} \pi(v_{\rho_\theta}) = \sum_{\theta \in \Theta \text{ s.t. } \theta \to \psi_s^j} \mu(\mathsf{S}^\theta) = 1.$$

- Clearly, π satisfies Φ^δ with probability 1 since $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ satisfies all instances of $\mathsf{Eq1}$-$\mathsf{Eq4}$, DEq, $\mathsf{E}(\Gamma)$, $\mathsf{D}(\Lambda)$, and a reasoning similar to the previous one can be applied.

- Finally, for each $r \in \{1, \ldots, s_j\}$, let ξ_r^j be a literal of the form $q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_n \cdot \mathsf{Pr}(\varphi_n) \bowtie q$, with $\bowtie \in \{\geq, <\}$. By (4.10), $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_n \cdot \mathsf{Pr}(\varphi_n) \bowtie q$, which, recalling once more (4.4), can be written as $q_1 \cdot \sum_{\theta \in \Theta \text{ st } \theta \to \varphi_1} \mu(\mathsf{S}^\theta) + \cdots + q_n \cdot \sum_{\theta \in \Theta \text{ st } \theta \to \varphi_n} \mu(\mathsf{S}^\theta) \bowtie q$ or even, recalling the definition of π, as

$$q_1 \cdot \sum_{\theta \in \Theta \text{ st } \theta \to \varphi_1} \pi(v_{\rho_\theta}) + \cdots + q_n \cdot \sum_{\theta \in \Theta \text{ st } \theta \to \varphi_n} \pi(v_{\rho_\theta}) \bowtie q.$$

  But note that for each $1 \leq i \leq n$, $\{v_{\rho_\theta} \mid \theta \in \Theta \text{ and } \theta \to \varphi_i\} = \left\{v \in \{0,1\}^{\mathcal{B}^\delta} \mid v(\varphi_i) = 1\right\}$ and we can write the previous inequality as

$$q_1 \cdot \sum_{v \in V^{\varphi_1}} \pi(v) + \cdots + q_n \cdot \sum_{v \in V^{\varphi_k}} \pi(v) \bowtie q,$$

  or, recalling the null assignment of probabilities for elements of $\{0,1\}^{\mathcal{B}^\delta} \setminus V$, as

$$q_1 \cdot \left(\sum_{v \in V^{\varphi_1}} \pi(v) + \sum_{v \in (\{0,1\}^{\mathcal{B}^\delta} \setminus V)^{\varphi_1}} \pi(v) \right) + \cdots + q_n \cdot \left(\sum_{v \in V^{\varphi_n}} \pi(v) + \sum_{v \in (\{0,1\}^{\mathcal{B}^\delta} \setminus V)^{\varphi_n}} \pi(v) \right) \bowtie q,$$

  i.e., $q_1 \cdot \sum_{v \in V^{\varphi_1}} \pi(v) + \cdots + q_n \cdot \sum_{v \in V^{\varphi_n}} \pi(v) \bowtie q$, and so π satisfies ξ_r^j.

For the reciprocal implication, assume that Algorithm 4.1 returns Sat and let V_1 be the set of valuations satisfying $\bigwedge\limits_{\mathsf{prop}_\psi \in H_j} \mathsf{prop}_\psi \wedge \neg \mathsf{prop}_{\varphi^j_\ell}$ for each $\ell \in \{1, \dots, k_j\}$, $V_1 = \{v_1, \dots, v_{k_j}\}$.

Then use Lemma 4.4.4 and let π be a probability distribution satisfying the $\mathsf{GenPSAT}$ instance (H_j, S_j) assigning a finite number of non-zero probabilities to valuations. Let V_2 denote the finite set of valuations for which π assigns non-zero probabilities. Please note that, since each propositional formula in H_j is satisfied with probability 1, and V_2 encompasses the valuations to which is assigned non-zero probability, it follows that all valuations in V_2 satisfy the propositional formulas in H_j. Let $V = V_1 \cup V_2$.

As we briefly explained, a model for δ is found by extending the signature F with new constants $c_{v,n}$, for each $n \in N$ and $v \in V$, that should keep hold the propositional behaviour into the logic. For this purpose, let F^* be an extension of F respecting (4.8). Then, consider the consistent set $M = \bigcup\limits_{i \in I} M_i \subseteq \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$ defined inductively on (4.9) and take a maximal consistent set Ξ^* extending M, whose existence is guaranteed by Lindenbaum's Lemma. Consider the F^*-algebra $\mathbb{A} = \mathbb{T}_{\mathrm{F}^*}(\varnothing)_{/\equiv^*}$ derived from the congruence relation defined by $t_1 \equiv^* t_2$ iff $\forall(t_1 \approx t_2) \in \Xi^*$. The interpretations for domain names are taken according to the maximal consistent set Ξ^*; $\mathrm{I}^{\mathbb{A}} : \mathcal{D} \to \wp(A)$ is defined as $\mathrm{I}^{\mathbb{A}}(\mathrm{D}) = \{[t]_{\equiv} \mid \forall(t \in \mathrm{D}) \in \Xi^* \text{ and } t \in T_{\mathrm{F}^*}(\varnothing)\}$ for each $\mathrm{D} \in \mathcal{D}$.

Afterwards, we translate valuations into outcomes, defining an outcome $\rho^v : N \to \mathbb{A}$ for each valuation $v \in V$ as $\rho^v(n) = [c_{v,n}]_{\equiv}$. Recall that $[\![\cdot]\!]^{\rho^v}_{\mathbb{A}} : T(N) \to \mathbb{A}$ is defined, accordingly, as $[\![t]\!]^{\rho^v}_{\mathbb{A}} = [[t]^{\tilde{n}}_{\tilde{c}_{v,n}}]_{\equiv}$, where $\tilde{n} = \mathsf{names}(t)$ and $\tilde{c}_{v,n}$ are the respective constants. The set of outcomes is defined as the union of two components, $\mathsf{S} = \mathsf{S}_1 \cup \mathsf{S}_2$, arising from V_1 and V_2, respectively: $\mathsf{S}_1 = \{\rho^v \mid v \in V_1\}$ and $\mathsf{S}_2 = \{\rho^v \mid v \in V_2\}$. Note that $\mathsf{S}_2 \neq \varnothing$ provided that $V_2 \neq \varnothing$ (see Lemma 4.4.4).

Regarding the probabilistic component, we first define a probability distribution over the outcomes in S and then use it to define the probability space. For the purpose, we import the probabilities on the propositional level through the probability distribution $\mathcal{P} : \mathsf{S} \to [0,1]$ defined by:

$$\begin{cases} \mathcal{P}(\rho^v) & = & \pi(v), & \text{for each } \rho^v \in \mathsf{S}_2, \\ \mathcal{P}(\rho) & = & 0, & \text{for } \rho \in \mathsf{S} \setminus \mathsf{S}_2. \end{cases}$$

A probability space $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$ is then defined using the discrete σ-algebra $\mathscr{A}$ over S, and the probability measure $\mu : \mathscr{A} \to [0,1]$ defined by $\mu(X) = \sum\limits_{\rho \in X} \mathcal{P}(\rho)$.

The conclusion that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ is actually an F-structure follows by the straightforward verification that μ is a probability measure.

- $\mathbb{A}$ **satisfies** Γ immediately by definition of M_0 and of the congruence relation $\equiv^*$.

- $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ **verifies** Λ**:** given $\left(t_1 \in \mathrm{D}_1, \ldots, t_{k_1} \in \mathrm{D}_{k_1} \to t'_1 \textcircled{$\in$} \mathrm{D}'_1, \ldots, t'_{k_2} \textcircled{$\in$} \mathrm{D}_{k_2}\right) \in \Lambda$ and $\sigma' \in A^X$, notice that σ' results from applying a substitution $\sigma \in T_{\mathrm{F}^*}(\varnothing)^X$ and then a quotient by $\equiv^*$. Assume that $[\![t_i]\!]^{\sigma'}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}_i)$ for each $1 \leq i \leq k_i$, which means that $[\sigma(t_i)]_{\equiv^*} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}_i)$ or, equivalently, $\forall(\sigma(t_i) \in \mathrm{D}_i) \in \Xi^*$. It means that $\forall(\sigma(t_1) \in \mathrm{D}_1 \wedge \cdots \wedge \sigma(t_{k_1}) \in \mathrm{D}_{k_1}) \in \Xi^*$, and from M_0 it follows that

$$\forall(\sigma(t'_1) \textcircled{$\in$} \mathrm{D}'_1 \vee \cdots \vee \sigma(t'_{k_2}) \textcircled{$\in$} \mathrm{D}'_{k_2}) \in \Xi^*.$$

  But Ξ^* is maximal consistent with respect to the deductive system $\mathcal{H}_{(\Gamma,\Lambda)}$ and $\sigma(t'_1)$,..., $\sigma(t'_{k_2})$ are nameless terms, so it follows that there exist $j \in \{1, \ldots, k_2\}$ such that $\forall(\sigma(t'_j) \textcircled{$\in$} \mathrm{D}'_j) \in \Xi^*$.

Once defined the F-structure $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$, let us check that for each $t, t_1, t_2 \in \mathsf{RelTerm}$, $\mathrm{D} \in \mathcal{D}$, $v \in V$, and denoting by $\tilde{n_{12}} = \mathsf{names}(t_1) \cup \mathsf{names}(t_2)$ and $\tilde{n} = \mathsf{names}(t)$,

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} \varphi \text{ iff } v\left(\mathsf{p}_\varphi\right) = 1 \text{ for each } \varphi \in \mathsf{Eq}(N) \cup \mathsf{DRes}(N):$$

$$\begin{aligned}
(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} t_1 \approx t_2 \quad & \text{iff} \quad [\![t_1]\!]^{\rho^v}_{\mathbb{A}} = [\![t_2]\!]^{\rho^v}_{\mathbb{A}} \\
& \text{iff} \quad [t_1]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \equiv [t_2]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \\
& \text{iff} \quad \forall([t_1]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \approx [t_2]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}}) \in \Xi^* \\
& \text{iff} \quad \forall([t_1]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}} \approx [t_2]^{\tilde{n_{12}}}_{\tilde{c}_{v,n_{12}}}) \in M \quad \text{(by Remark 4.4.7)} \\
& \text{iff} \quad v\left(\mathsf{p}_{t_1 \approx t_2}\right) = 1 \\
(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} t \in \mathrm{D} \quad & \text{iff} \quad [\![t]\!]^{\rho^v}_{\mathbb{A}} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}) \\
& \text{iff} \quad [[t]^{\tilde{n}}_{\tilde{c}_{v,n}}]_{\equiv} \in \mathrm{I}^{\mathbb{A}}(\mathrm{D}) \\
& \text{iff} \quad \forall([t]^{\tilde{n}}_{\tilde{c}_{v,n}} \in \mathrm{D}) \in \Xi^* \\
& \text{iff} \quad \forall([t]^{\tilde{n}}_{\tilde{c}_{v,n}} \in \mathrm{D}) \in M \quad \text{(by Remark 4.4.7)} \\
& \text{iff} \quad v\left(\mathsf{p}_{t \in \mathrm{D}}\right) = 1
\end{aligned}$$

By induction we easily conclude that for each $\varphi \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and $v \in V$,

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} \varphi \text{ iff } v\left(\mathsf{prop}_\varphi\right) = 1. \tag{4.12}$$

It remains to prove that, actually, $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ is a model for δ. For that, recall that δ is given in DNF by $\bigvee_{j=1}^{m}\left(\forall\psi^j_1 \wedge \ldots \wedge \forall\psi^j_{n_j} \wedge \neg\forall\varphi^j_1 \wedge \ldots \wedge \neg\forall\varphi^j_{k_j} \wedge \xi^j_1 \wedge \ldots \wedge \xi^j_{s_j}\right)$ and let us check that we have a model for the j^{th} disjunct.

- For each $s \in \{1, \ldots, n_j\}$ and $v \in V$, $\psi^j_s \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and $v\left(\mathsf{prop}_{\psi^j_s}\right) = 1$, so, by (4.12), $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} \psi^j_s$, and it follows that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathsf{S}) \Vdash \forall\psi^j_s$.

- For each $\ell \in \{1, \ldots, k_j\}$, $\varphi_\ell^j \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$, and there exists $v_\ell \in V_1$ such that $v_\ell\left(\mathsf{prop}_{\varphi_\ell^j}\right) = 0$. Then $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^{v_\ell} \not\Vdash_{\mathsf{loc}} \varphi_\ell^j$ and so $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathsf{S}) \Vdash \neg\forall\varphi_\ell^j$.

- Finally, since π satisfies the $\mathsf{GenPSAT}$ instance (H_j, S_j), given $r \in \{1, \ldots, s_j\}$, in particular, π satisfies ξ_r^j. Let ξ_r^j be a probabilistic formula of the form $q_1 \cdot \mathsf{Pr}(\varphi_1) + \cdots + q_n \cdot \mathsf{Pr}(\varphi_n) \bowtie q$, with $\bowtie \in \{\geq, <\}$, it follows that
$$q_1 \cdot \sum_{v \in V^{\varphi_1}} \pi(v) + \cdots + q_n \cdot \sum_{v \in V^{\varphi_n}} \pi(v) \bowtie q$$
which, by (4.12) and by definition of $\mathcal{P}$, is equivalent to
$$q_1 \cdot \sum_{\rho^v \in \mathsf{S}^{\varphi_1}} \mathcal{P}(\rho^v) + \cdots + q_n \cdot \sum_{\rho^v \in \mathsf{S}^{\varphi_n}} \mathcal{P}(\rho^v) \bowtie q$$
that can be written as $q_1 \cdot \mu(\mathsf{S}^{\varphi_1}) + \cdots + q_n \cdot \mu(\mathsf{S}^{\varphi_n}) \bowtie q$ and is exactly what we want: $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \xi_r^j$. □

Notice that, given a global formula $\delta \in \mathsf{Glob}$ written in DNF, Algorithm 4.1 makes a polynomial number of calls to a couple of oracles: a SAT oracle and a $\mathsf{GenPSAT}$ oracle. It results in high complexity costs. The $\mathsf{DNFSAT\text{-}DEqPrL}$ solver that we present is in P^{NP}. Furthermore, rewriting a given global formula into disjunctive normal form can lead to an explosion on the length of the formula. For these reasons, even though Algorithm 4.1 is intuitive and easily explained, it does not constitute the most efficient way to decide $\mathsf{SAT\text{-}DEqPrL}$.

$\mathsf{CNFSAT\text{-}DEqPrL}$ problem

Despite the high complexity costs that we achieved with the presented $\mathsf{DNFSAT\text{-}DEqPrL}$ algorithm, it has given us a valuable hint in the search for a more efficient satisfiability algorithm. As we already know, we should focus on the problem whose input is given in CNF.

The $\mathsf{CNFSAT\text{-}DEqPrL}$ problem consists in deciding the existence of a model for a global formula $\delta \in \mathsf{Glob}$ given in conjunctive normal form. We analyze the $\mathsf{CNFSAT\text{-}DEqPrL}$ problem inspired on the developments for $\mathsf{GenPSAT}$ presented in Chapter 3 and explore a polynomial reduction to the Satisfiability Modulo Theories (SMT) with respect to the theory of quantifier--free linear arithmetic over the integers and reals ($\mathsf{QF\_LIRA}$) [20].

On the details of the input formula: Assume that we are given a global formula $\delta \in \mathsf{Glob}$ given in CNF by $\bigwedge_{j=1}^{m} \bigvee_{i=1}^{n_j} \delta_i^j$. Since each conjunct of δ is a disjunction of literals in $\forall\mathsf{Loc} \cup \neg\forall\mathsf{Loc} \cup \mathsf{Prob} \cup \neg\mathsf{Prob}$, we can rewrite it as:
$$\bigwedge_{j=1}^{m} (\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j \vee \xi_1^j \vee \ldots \vee \xi_{s_j}^j)\,, \tag{4.13}$$
where, for each $r \in \{1, \ldots, s_j\}$, the probabilistic literal ξ_r^j is assumed to take the following form: $q_{(r,j,1)} \cdot \mathsf{Pr}(\varphi_{(r,j,1)}) + \cdots + q_{(r,j,\ell_r^j)} \cdot \mathsf{Pr}(\varphi_{(r,j,\ell_r^j)}) \bowtie_r^j q_{(r,j)}$, with $\bowtie_r^j \in \{\geq, <\}$.

Witnessing existential literals: Following the same motivation to address the need of witnesses for existential literals as for $\mathsf{CNFSAT\text{-}EqCL}$ in Section 2.4, we need, at least, as many copies of $\mathcal{B}^{\delta}$ (4.6) as the number of existential formulas $\neg\forall\mathsf{Loc}$ occurring in δ. In its description, δ counts with $\sum_{j=1}^{m} k_j$ literals of $\neg\forall\mathsf{Loc}$, so the final set of propositional symbols should contain all the required copies, $\bigcup\limits_{j'=1}^{m}\bigcup\limits_{\ell'=1}^{k_{j'}}\{\mathsf{p}^{[j',\ell']} \mid \mathsf{p}\in\mathcal{B}^{\delta}\}$. When $k_{j'}=0$, $\bigcup\limits_{\ell'=1}^{k_{j'}}\{\mathsf{p}^{[j',\ell']} \mid \mathsf{p}\in\mathcal{B}^{\delta}\}$ represents the empty set.

Probability also deserves a witness: As we know, the probabilistic feature envisage a probability distribution over the set of valuations. In this sense, we should not limit our valuations to strictly represent witnesses for existential literals. Hence, we further need to consider an extra copy of $\mathcal{B}^{\delta}$.

Interpreting $\forall\mathsf{Loc}$ and $\neg\forall\mathsf{Loc}$ in the propositional context: Satisfying an element of the form $\forall\varphi$ imposes that φ must be verified in all possible outcome, whereas satisfying a formula as $\neg\forall\varphi$ requires that at least one possible outcome satisfies $\neg\varphi$. Therefore, our reduction to the propositional context must carry this sensitivity. In this way, the satisfiability of those literals is tested using several labeled copies of propositional variables (one copy for each literal of the form $\neg\forall\mathsf{Loc}$ plus the original copy), as if they had embedded several valuations. The *labels* are extended from the propositional variables to the propositional formulas as expected.

Perfect tuning with $\mathsf{GenPSAT}$: Prompted by the inclusion of SAT in $\mathsf{GenPSAT}$, the satisfiability of propositional formulas (representing literals in $\forall\mathsf{Loc}$) is tested by assigning to it probability 1. Accordingly, and inspired on the $\mathsf{GenPSAT}$ normal forms (see Chapter 3), we realize that the probabilistic (propositional) formulas to be tested should be *atomic.* For this purpose, we shall replace the propositional formulas occurring inside probabilistic (propositional) formulas by *ghost* propositional symbols. The existential literals are not supposed to influence probabilities (they have their own witnesses), so we discard them for a moment. Let us collect in $\mathfrak{SLoc}$ all the appropriate local formulas, suggested by δ:

$$\mathfrak{SLoc} = \bigcup_{j=1}^{m}\left(\{\psi_1^j,\dots,\psi_{n_j}^j\}\cup\bigcup_{r=1}^{s_j}\{\varphi_{(r,j,1)},\dots,\varphi_{(r,j,\ell_r^j)}\}\right),$$

and in $\mathfrak{G}$ the corresponding propositional symbols:

$$\mathfrak{G} = \bigcup_{j=1}^{m}\left(\{\mathfrak{p}_{\psi_1^j},\dots,\mathfrak{p}_{\psi_{n_j}^j}\}\cup\bigcup_{r=1}^{s_j}\{\mathfrak{p}_{\varphi_{(r,j,1)}},\dots,\mathfrak{p}_{\varphi_{(r,j,\ell_r^j)}}\}\right).$$

Furthermore, for each $\psi\in\mathfrak{SLoc}$, the $[0,1]$-variable α_{ψ} is intended to represent the probability of ψ.

Incorporating Φ^δ: As we have already seen, in order to obtain a correct translation into the propositional context, we should impose the requirements collected in Φ^δ (4.7). For this purpose, all the considered copies of $\mathcal{B}^\delta$ must verify those restrictions (with probability 1). And so, we should keep a special propositional *ghost* symbol for this purpose, $\mathfrak{p}_\phi$, and a variable to represent its probability, α_ϕ.

The translation to $\mathsf{QF\_LIRA}$: All these things considered, let

$$\widetilde{\mathcal{B}} = \mathcal{B}^\delta \cup \bigcup_{j'=1}^{m} \bigcup_{\ell'=1}^{k_{j'}} \{\mathsf{p}^{[j',\ell']} \mid \mathsf{p} \in \mathcal{B}^\delta\} \cup \mathfrak{G} \cup \{\mathfrak{p}_\phi\}$$

represent the set of propositional symbols for our problem and denote by M the number of elements of $\mathfrak{G} \cup \{\mathfrak{p}_\phi\}$, $M \leq \sum\limits_{j=1}^{m} \left(n_j + \sum\limits_{r=1}^{s_j} \ell_r^j \right) + 1$. For ease of notation, let

$$\nu : \mathfrak{SLoc} \cup \{\phi\} \to \{1, \ldots, M\}$$

represent a bijection from the $\mathfrak{SLoc}$ coupled with the symbol ϕ to the set $\{1, \ldots, M\}$ such that $\nu(\phi) = M$. Note that the inverse bijection ν^{-1} is such that $\nu^{-1}(\{1, \ldots, M\}) = \mathfrak{SLoc} \cup \{\phi\}$.

Inspired in Subsection 3.3.2, let $H = [h_{ij}]$ denote a (still unknown) matrix of size $M \times (M+1)$ whose columns represent the valuations over $\widetilde{\mathcal{B}}$ evaluated on each propositional (ghost) symbol of $\mathfrak{G} \cup \{\mathfrak{p}_\phi\}$, i.e., $h_{ik} = v_k\left(\mathfrak{p}_{\nu^{-1}(i)}\right)$ for each $1 \leq i \leq M$ and $1 \leq k \leq M+1$. The $(M+1)$-vector $\pi = [\pi_k]$ represents a probability distribution over $\{v_1, \ldots, v_{M+1}\}$. As we already mentioned, α_ψ represents the probability of each $\psi \in \mathfrak{SLoc}$ and α_ϕ aims to represent the probability of Φ^δ.

In order to model all the possible valuations $\{v_1, ..., v_{M+1}\}$, we consider $M+1$ copies of $\widetilde{\mathcal{B}}$:

$$\mathcal{B}^* = \bigcup_{k=1}^{M+1} \{{}^{(k)}\mathsf{p} \mid \mathsf{p} \in \widetilde{\mathcal{B}}\}.$$

Given $\mathcal{F} \subseteq \widetilde{\mathcal{B}}$, we denote by ${}^{(k)}\mathcal{F}$ the set $\{{}^{(k)}\mathsf{p} \mid \mathsf{p} \in \mathcal{F}\}$.

The idea is to test the satisfiability of δ through the assertion

$$\textbf{(prob)} \quad \bigwedge_{j=1}^{m} \left(\bigvee_{s=1}^{n_j} \left(\alpha_{\psi_s^j} = 1 \right) \vee \bigvee_{\ell=1}^{k_j} \left(\bigvee_{k=1}^{M+1} {}^{(k)}\mathsf{prop}_{\neg\varphi_\ell^j}^{[j,\ell]} \right) \vee \bigvee_{r=1}^{s_j} \left(\sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \alpha_{\varphi_{(r,j,s)}} \bowtie_r^j q_{(r,j)} \right) \right)$$

subject to the additional assertions:

$$\textbf{(prop\_pos)} \quad \bigwedge_{k=1}^{M+1} \left({}^{(k)}\mathfrak{p}_{\psi_s^j} \leftrightarrow \left(\bigwedge_{j'=1}^{m} \bigwedge_{\ell'=1}^{k_{j'}} {}^{(k)}\mathsf{prop}_{\psi_s^j}^{[j',\ell']} \wedge {}^{(k)}\mathsf{prop}_{\psi_s^j} \right) \right), \text{ for each } \psi_s^j \in \mathfrak{SLoc};$$

$$\textbf{(prop\_prob)} \quad \bigwedge_{k=1}^{M+1} \left({}^{(k)}\mathfrak{p}_{\varphi_{(r,j,s)}} \leftrightarrow {}^{(k)}\mathsf{prop}_{\varphi_{(r,j,s)}} \right), \text{ for each } \varphi_{(r,j,s)} \in \mathfrak{SLoc};$$

(prop_phi) $\bigwedge_{k=1}^{M+1}\left({}^{(k)}\mathfrak{p}_{\phi} \leftrightarrow \bigwedge_{\phi\in\Phi^{\delta}}\left(\bigwedge_{j'=1}^{m}\bigwedge_{\ell'=1}^{k_{j'}} {}^{(k)}\phi^{[j',\ell']} \wedge {}^{(k)}\phi\right)\right)$;

(prob_phi) $(\alpha_{\phi} = 1)$;

(val1) $\left(\sum_{k=1}^{M+1} b_{ik} = \alpha_{\nu^{-1}(i)}\right)$, for each $i \in \{1,\ldots,M\}$;

(val2) $((0 \le b_{ik} \le h_{ik}) \wedge (h_{ik} - 1 + \pi_k \le b_{ik} \le \pi_k))$, for each $i \in \{1,...,M\}$ and $k \in \{1,...,M+1\}$;

(cons) $\left(h_{ik} = 1 \leftrightarrow {}^{(k)}\mathfrak{p}_{\nu^{-1}(i)}\right)$, for each $i \in \{1,\ldots,M\}$ and $k \in \{1,\ldots,M+1\}$;

(sums1) $\left(\sum_{i=1}^{M+1} \pi_k = 1\right)$.

So far we have introduced:

- 1 assertion (prob);
- $\sum_{j=1}^{m} n_j$ assertions (prop_pos);
- $\sum_{j=1}^{m}\left(\sum_{r=1}^{s_j} \ell_r^j\right)$ assertions (prop_prob);
- 1 assertions (prop_phi);
- M assertions (val1);
- $M \times (M+1)$ assertions (val2);
- $M \times (M+1)$ assertions (cons);
- 1 assertions (sums1).

Hence, we have $\mathcal{O}(M + M \times (M+1))$ assertions, each of polynomial size on the length of δ, over $M \times (M+1)$ binary variables h_{ij}, $M \times (M+1)$ real variables b_{ij}, M real variables $0 \le \alpha_{\psi} \le 1$, $(M+1)$ real variables $0 \le \pi_k \le 1$ and $(M+1) \cdot \left(|\mathcal{B}_{\delta}| + |\mathcal{B}_{\delta}| \cdot \sum_{j=1}^{m} k_{j'} + M\right)$ propositional variables, where $M = \sum_{j=1}^{m}\left(n_j + \sum_{r=1}^{s_j} \ell_r^j\right) + 1$. Because of this, the presented translation to QF_LIRA is polynomial.

The solver: We test the satisfiability of δ by translating it into a QF_LIRA problem and then solving the latter appropriately. The procedure is presented in Algorithm 4.2.

Given a global formula $\delta \in$ Glob written in conjunctive normal form, the CNFSAT-DEqPrL solver tests the satisfiability of δ by reduction to a QF_LIRA problem with polynomial size on the length of δ. Note that we have fixed a convergent equational theory Γ and a set of domain clauses Λ with the subterm property, so that the sets of propositional formulas $\mathcal{B}^{\delta}$

and Φ^δ are well defined and have polynomial size on the length of δ. The procedure consists in initializing an empty QF_LIRA problem and then use the following auxiliary procedures:

- assert introduces an assertion into the QF_LIRA problem;
- qf_lira_solver returns Sat or Unsat depending on whether the problem is satisfiable or not.

When the resulting QF_LIRA problem is satisfiable, we conclude that δ is also satisfiable.

Algorithm 4.2 CNFSAT-DEqPrL solver based on SMT – QF_LIRA

```
procedure CNFSATDEqPrL
    input: CNF global formula $\delta$: $\bigwedge_{j=1}^{m} \left(\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j \vee \xi_1^j \vee \ldots \vee \xi_{s_j}^j\right)$
    output: Sat or Unsat depending on whether $\delta$ is satisfiable or not
    assume: $M := \sum_{j=1}^{m} (n_j + \sum_{r=1}^{s_j} \ell_r^j) + 1$
            $\nu : \mathfrak{SLoc} \cup \{\phi\} \to \{1, \ldots, M\}$ is a bijection
    declare: propositional variables: $\bigcup_{k=1}^{M+1} \left( {}^{(k)}\mathcal{B}^\delta \cup \bigcup_{j'=1}^{m} \bigcup_{\ell'=1}^{k_{j'}} \{ {}^{(k)}\mathsf{p}^{[j',\ell']} \mid \mathsf{p} \in \mathcal{B}^\delta \} \cup {}^{(k)}\mathfrak{G} \cup \{ {}^{(k)}\mathfrak{p}_\phi \} \right)$
             binary variables: $h_{ik}$, for $i \in \{1, \ldots, M\}$, $k \in \{1, \ldots, M+1\}$
             $[0,1]$-variables: $\alpha_{\nu^{-1}(i)}, \pi_k, b_{ik}$, for $i \in \{1, \ldots, M\}$, $k \in \{1, \ldots, M+1\}$
    for $j = 1$ to $m$ do
        assert $\left( \bigwedge_{k=1}^{M+1} \bigwedge_{s=1}^{n_j} \left( {}^{(k)}\mathfrak{p}_{\psi_s^j} \leftrightarrow \left( \bigwedge_{j'=1}^{m} \bigwedge_{\ell'=1}^{k_{j'}} {}^{(k)}\mathsf{prop}_{\psi_s^j}^{[j',\ell']} \wedge {}^{(k)}\mathsf{prop}_{\psi_s^j} \right) \right) \right)$    ▷ (prop_pos)
        assert $\left( \bigwedge_{k=1}^{M+1} \bigwedge_{r=1}^{s_j} \bigwedge_{s=1}^{\ell_r^j} \left( {}^{(k)}\mathfrak{p}_{\varphi_{(r,j,s)}} \leftrightarrow {}^{(k)}\mathsf{prop}_{\varphi_{(r,j,s)}} \right) \right)$    ▷ (prop_prob)
    for $i = 1$ to $M$ do
        assert $\left( \sum_{k=1}^{M+1} b_{ik} = \alpha_{\nu^{-1}(i)} \right)$    ▷ (val1)
        for $k = 1$ to $M + 1$ do
            assert $((0 \le b_{ik} \le h_{ik}) \wedge (h_{ik} - 1 + \pi_k \le b_{ik} \le \pi_k))$    ▷ (val2)
            assert $\left( h_{ik} = 1 \leftrightarrow {}^{(k)}\mathfrak{p}_{\nu^{-1}(i)} \right)$    ▷ (cons)
    assert $\left( \bigwedge_{j=1}^{m} \left( \bigvee_{s=1}^{n_j} \left( \alpha_{\psi_s^j} = 1 \right) \vee \bigvee_{\ell=1}^{k_j} \left( \bigvee_{k=1}^{M+1} {}^{(k)}\mathsf{prop}_{\neg\varphi_\ell^j}^{[j,\ell]} \right) \vee \bigvee_{r=1}^{s_j} \left( \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \alpha_{\varphi_{(r,j,s)}} \bowtie_r^j q_{(r,j)} \right) \right) \right)$    ▷ (prob)
    assert $\left( \bigwedge_{k=1}^{M+1} \left( {}^{(k)}\mathfrak{p}_\phi \leftrightarrow \bigwedge_{\phi \in \Phi^\delta} \left( \bigwedge_{j'=1}^{m} \bigwedge_{\ell'=1}^{k_{j'}} {}^{(k)}\phi^{[j',\ell']} \wedge {}^{(k)}\phi \right) \right) \right)$    ▷ (prop_phi)
    assert $\left( \alpha_\phi = 1 \right)$    ▷ (prob_phi)
    assert $\left( \sum_{k=1}^{M+1} \pi_k = 1 \right)$    ▷ (sums1)
    return qf_lira_solver()    ▷ return Sat if the assertions are satisfiable, Unsat otherwise
```

For the sake of illustration, we now use this algorithm to decide whether a global formula is satisfiable or not. Later on, we will illustrate the importance of each variable of the $\mathsf{QF\_LIRA}$ problem in the construction of a model.

Example 4.4.9. Recall Example 4.1.1 and consider the signature $\mathrm{F}^{\mathrm{xor}}$, the equational theory Γ^{xor} and the axiomatization Λ^{xor}. Let us test the satisfiability of the CNF global formula:

$$\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \cdot \mathsf{Pr}(n \in \mathsf{even}) \wedge \forall (n \in \mathsf{even}) \wedge \left(\neg \mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \vee \neg \forall \mathsf{suc}(n) \in \mathsf{odd} \right),$$

with $n \in \mathbb{N}$. We start by noting that $\mathsf{RelTerm}^{\delta} = \{n, \mathsf{zero}, \mathsf{suc}(n)\}$ and defining Φ^{δ}. The parameters that come into play when the formula is seen in the CNF form (4.13), $\bigwedge_{j=1}^{m} \left(\forall \psi_1^j \vee \ldots \vee \forall \psi_{n_j}^j \vee \neg \forall \varphi_1^j \vee \ldots \vee \neg \forall \varphi_{k_j}^j \vee \xi_1^j \vee \ldots \vee \xi_{s_j}^j \right)$, are:

$$\begin{array}{llllll} j = 1: & n_1 = 0 & k_1 = 0 & s_1 = 1 & \ell_1^1 = 2 \\ j = 2: & n_2 = 1 & k_2 = 0 & s_2 = 0 & \\ j = 3: & n_3 = 0 & k_3 = 1 & s_3 = 1 & \ell_3^1 = 1 \end{array}$$

Note that

$$\mathfrak{SLoc} = \{n \approx \mathsf{zero}, n \in \mathsf{even}\}$$

and consider the bijection $\nu : \mathfrak{SLoc} \cup \{\phi\} \to \{1, 2, 3\}$ such that $\nu(n \approx \mathsf{zero}) = 1$, $\nu(n \in \mathsf{even}) = 2$, $\nu(\phi) = 3$.

To clarify the following observations, let us remind all the relevant assertions that will allow us to draw a conclusion about the satisfiability of the formula:

(prob) $\bigwedge_{j=1}^{m} \left(\bigvee_{s=1}^{n_j} \left(\alpha_{\psi_s^j} = 1 \right) \vee \bigvee_{\ell=1}^{k_j} \left(\bigvee_{k=1}^{M+1} {}^{(k)}\mathsf{prop}_{\neg \varphi_\ell^j}^{[j,\ell]} \right) \vee \bigvee_{r=1}^{s_j} \left(\sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \alpha_{\varphi_{(r,j,s)}} \bowtie_r^j q_{(r,j)} \right) \right)$

(prop_pos) $\bigwedge_{k=1}^{M+1} \left({}^{(k)}\mathfrak{p}_{\psi_s^j} \leftrightarrow \left(\bigwedge_{j'=1}^{m} \bigwedge_{\ell'=1}^{k_{j'}} {}^{(k)}\mathsf{prop}_{\psi_s^j}^{[j',\ell']} \wedge {}^{(k)}\mathsf{prop}_{\psi_s^j} \right) \right)$, for each $\psi_s^j \in \mathfrak{SLoc}$;

(val1) $\left(\sum_{k=1}^{M+1} b_{ik} = \alpha_{\nu^{-1}(i)} \right)$, for each $i \in \{1, \ldots, M\}$;

(val2) $((0 \leq b_{ik} \leq h_{ik}) \wedge (h_{ik} - 1 + \pi_k \leq b_{ik} \leq \pi_k))$, for each $i \in \{1, ..., M\}$ and $k \in \{1, ..., M+1\}$;

(cons) $\left(h_{ik} = 1 \leftrightarrow {}^{(k)}\mathfrak{p}_{\nu^{-1}(i)} \right)$, for each $i \in \{1, \ldots, M\}$ and $k \in \{1, \ldots, M+1\}$.

For the given formula, the assertion (prob) reads like

$$\left(\alpha_{n \approx \mathsf{zero}} \leq \frac{2}{3} \cdot \alpha_{n \in \mathsf{even}} \right) \wedge \left(\alpha_{n \in \mathsf{even}} = 1 \right) \wedge \left(\alpha_{n \approx zero} > \frac{2}{3} \ \vee \ \bigvee_{k=1}^{4} {}^{(k)}\mathsf{prop}_{\neg \mathsf{suc}(n) \in \mathsf{odd}}^{[3,1]} \right),$$

which together with the remaining assertions carefully described in Algorithm 4.2 is unsatisfiable. To check that, assume that it would have a solution (denoted by x^* for each variable x) and let us derive a contradiction.

Begin noting that by $(\mathsf{val1})$, $b_{\nu^{-1}(n\in\mathsf{even}),k}$ ranges in the interval $[0,\pi_k]$ for each $k\in\{1,\ldots,5\}$. Once $\alpha^*_{n\in\mathsf{even}}=1$, then every $b_{\nu^{-1}(n\in\mathsf{even}),k}$ should coincide with π_k and, by $(\mathsf{val2})$, $h^*_{\nu^{-1}(n\in\mathsf{even}),k}=1$ for every $k\in\{1,2,3,4,5\}$. Then, by (cons), ${}^{(k)}\mathfrak{p}_{n\in\mathsf{even}}$ holds. But, by $(\mathsf{prop\_pos})$ this means that for each k, ${}^{(k)}\mathsf{prop}^{[3,1]}_{n\in\mathsf{even}}\wedge{}^{(k)}\mathsf{prop}_{n\in\mathsf{even}}$ also holds.

Observing that $(n\in\mathsf{even}\to\mathsf{suc}(n)\in\mathsf{odd})\in\Phi^\delta$, it follows that for each k,

$$ {}^{(k)}\mathsf{prop}^{[3,1]}_{\mathsf{suc}(n)\in\mathsf{odd}}\wedge{}^{(k)}\mathsf{prop}_{\mathsf{suc}(n)\in\mathsf{odd}}\ \text{ holds.} \tag{4.14} $$

Then, we have that ${}^{(k)}\mathsf{prop}^{[3,1]}_{\neg\mathsf{suc}(n)\in\mathsf{odd}}$ does not hold for every k. On the other hand, since $\alpha^*_{n\approx\mathsf{zero}}\leq\frac{2}{3}$, there is no way for the last conjunct to hold and we conclude that the formula is unsatisfiable. △

Now that we checked how to apply the procedure, let us prove its correctness.

Lemma 4.4.10. *If* Γ *is a convergent equational theory and* Λ *is a set of domain clauses with the subterm property, a global formula* $\delta\in\mathsf{Glob}$ *in CNF is satisfiable iff Algorithm 4.2 returns* Sat*.*

Proof of Lemma 4.4.10. Assume that Γ is a convergent equational theory, Λ is a set of domain clauses with the subterm property and let $\delta\in\mathsf{Glob}$ be any global formula in CNF. To prove the direct implication, consider an F-structure $(\mathbb{A},\mathrm{I}^{\mathbb{A}},\mathbb{P})$ with $\mathbb{P}=(\mathsf{S},\mathscr{A},\mu)$ and satisfying δ: $(\mathbb{A},\mathrm{I}^{\mathbb{A}},\mathbb{P})\Vdash\bigwedge_{j=1}^{m}\bigvee_{i=1}^{n_j}\delta^j_i$. Since each δ^j_i is a literal, we can rewrite it as

$$ (\mathbb{A},\mathrm{I}^{\mathbb{A}},\mathbb{P})\Vdash\bigwedge_{j=1}^{m}(\forall\psi^j_1\vee\ldots\vee\forall\psi^j_{n_j}\vee\neg\forall\varphi^j_1\vee\ldots\vee\neg\forall\varphi^j_{k_j}\vee\xi^j_1\vee\ldots\vee\xi^j_{s_j}). \tag{4.15} $$

Recall the construction carried out in the proof of completeness of DEqPrL, and which was revisited in the proofs of Lemmas 4.4.2 and 4.4.4. Let Ω be the set of the local formulas occurring inside the probabilistic literals occurring in δ, $\Omega=\bigcup_{j=1}^{m}\mathsf{lnPr}(\xi^j_1)\cup\ldots\cup\mathsf{lnPr}(\xi^j_{s_j})$, and define the suitable atoms $\Theta=\left\{\bigwedge_{\gamma\in\Upsilon}\gamma\wedge\bigwedge_{\omega\in\Omega\setminus\Upsilon}\neg\omega\mid\Upsilon\subseteq\Omega\right\}$. We assign an outcome $\rho_\theta\in\mathsf{S}$ for each $\theta\in\Theta$ such that $\mathsf{S}^\theta\neq\varnothing$ and consider the set of valuations $V_1\subseteq\{0,1\}^{\mathcal{B}^\delta}$ arising from this construction:

$$ V_1=\{v_{\rho_\theta}\mid\theta\in\Theta\text{ and }\mathsf{S}^\theta\neq\varnothing\}. $$

Notice that $V_1\neq\varnothing$ due to the suitable construction of Θ, which implies that $\bigcup_{\theta\in\Theta}\mathsf{S}^\theta=\mathsf{S}$.

Furthermore, let us collect in V_2 one valuation that attests the satisfiability of each existential literal, when it does exist, and a valuation arising from any other outcome in S when this disjunct is not satisfiable:

$$ V_2=\bigcup_{j=1}^{m}\bigcup_{\ell=1}^{k_j}\left(\left\{v_{\rho^{\varphi^j_\ell}}\mid(\mathbb{A},\mathrm{I}^{\mathbb{A}}),\rho^{\varphi^j_\ell}\Vdash_{\mathsf{loc}}\neg\varphi^j_\ell\right\}\cup\left\{v_\rho\mid\rho\in\mathsf{S}\text{ and }(\mathbb{A},\mathrm{I}^{\mathbb{A}},\mathbb{P})\Vdash\forall\varphi^j_\ell\right\}\right). $$

Consider the set of valuations over the wider set of propositional symbols $\widetilde{\mathcal{B}}$,

$$V_1^* = \{v_{\rho_\theta}^* \mid \theta \in \Theta, \mathsf{S}^\theta \neq \varnothing\}$$

where, for each $\theta \in \Theta$ with $\mathsf{S}^\theta \neq \varnothing$, $v_{\rho_\theta}^* : \widetilde{\mathcal{B}} \to \{0,1\}$ is the valuation based on v_{ρ_θ} that incorporates the information of every valuation that attests the satisfiability of existential literals and is collected in V_2. Thus, $v_{\rho_\theta}^*$ is defined by:

$$\begin{cases} v_{\rho_\theta}^*(\mathsf{p}) \ = \ v_{\rho_\theta}(\mathsf{p}) \\ v_{\rho_\theta}^*\left(\mathsf{p}^{[j,\ell]}\right) \ = \ v_{\rho^{\varphi_\ell^j}}(\mathsf{p}) \\ v_{\rho_\theta}^*\left(\mathfrak{p}_{\varphi_{(r,j,s)}}\right) \ = \ v_{\rho_\theta}\left(\mathsf{prop}_{\varphi_{(r,j,s)}}\right) \\ v_{\rho_\theta}^*\left(\mathfrak{p}_{\psi_s^j}\right) \ = \ \prod\limits_{j=1}^{m}\prod\limits_{\ell=1}^{k_j} v_{\rho^{\varphi_\ell^j}}\left(\mathsf{prop}_{\psi_s^j}\right) \cdot v_{\rho_\theta}\left(\mathsf{prop}_{\psi_s^j}\right) \\ v_{\rho_\theta}^*\left(\mathfrak{p}_\phi\right) \ = \ \prod\limits_{\phi\in\Phi}\prod\limits_{j=1}^{m}\prod\limits_{\ell=1}^{k_j} v_{\rho^{\varphi_\ell^j}}(\phi) \cdot v_{\rho_\theta}(\phi) \end{cases}$$

A probability distribution $\pi : \{0,1\}^{\widetilde{\mathcal{B}}} \to [0,1]$ is then defined using μ,

$$\begin{cases} \pi(v_{\rho_\theta}^*) & = & \mu(\mathsf{S}^\theta)\,, & \text{for each } \theta \in \Theta \text{ such that } \mathsf{S}^\theta \neq \varnothing \\ \pi(v) & = & 0\,, & \text{for the remaining } v \in \{0,1\}^{\widetilde{\mathcal{B}}} \smallsetminus V_1^*. \end{cases}$$

We observe that π is obviously a probability distribution over $\{0,1\}^{\widetilde{\mathcal{B}}}$ since it is always non-negative and, recalling that $\bigcup\limits_{\theta\in\Theta} \mathsf{S}^\theta = \mathsf{S}$ and $\mathsf{S}^{\theta_1} \cap \mathsf{S}^{\theta_2} = \varnothing$ whenever $\theta_1 \neq \theta_2$, π sums up to 1:

$$\sum_{v\in\{0,1\}^{\widetilde{\mathcal{B}}}} \pi(v) = \sum_{\theta\in\Theta} \pi(v_{\rho_\theta}^*) = \sum_{\theta\in\Theta} \mu(\mathsf{S}^\theta) = 1.$$

It remains to check that the assertions are satisfied. To check that (val1) and (val2) are satisfied, let us begin by establishing the probabilities of each propositional symbol in $\mathfrak{G}$: given $i \in \{1,\ldots,M\}$,

$$\alpha_{\nu^{-1}(i)}^* = \sum_{\substack{v^*\in V_1^* \\ v^*\left(\mathfrak{p}_{\nu^{-1}(i)}\right)=1}} \pi(v^*).$$

Then, consider the system of linear inequalities

$$\begin{cases} \overline{H}\,\overline{\pi} = \alpha^* \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} \tag{4.16}$$

where $\overline{H}$ is a $M \times K$ matrix of binary variables whose columns represent the valuations of V_1 evaluated on each propositional (ghost) symbol of $\mathfrak{G} \cup \{\mathfrak{p}_\phi\}$. Notice that, since Ω has $\sum_{j=1}^{m} \sum_{k=1}^{s_j} \ell_r^j$ local formulas, V_1 has at most $2^{\sum_{j=1}^{m} \sum_{k=1}^{s_j} \ell_r^j}$ valuations. K represents the number of elements of V_1. Furthermore, $\overline{\pi}$ is a K-vector of $[0,1]$-variables that aims to represent a probability distribution, and α^* is the $(M+1)$-vector of reals $\alpha^* = [\alpha^*_{\nu^{-1}(i)}]$.

A solution for (4.16) is found by considering an enumeration for the valuations in V_1 and then defining $H^* = [h^*_{ik}]$ with

$$h^*_{ik} = v^*_k \left(\mathfrak{p}_{\nu^{-1}(i)}\right)$$

and the K-vector $\pi^* = [\pi_k]$ with

$$\pi^*_k = \pi(v^*_k).$$

Using Lemma 1.2.3, we also conclude that there exists a solution for (4.16) with at most $M+1$ positive entries, which obviously satisfies $(\mathsf{sums1})$ and, considering

$$b^*_{ik} = h^*_{ik} \cdot \pi^*_k,$$

also satisfies $(\mathsf{val1})$ and $(\mathsf{val2})$. Assume without loss of generality that the non-negative entries referred by Lemma 1.2.3 are within the first $M+1$ columns of H^* and, from now on, let H^* and π^* represent only the first $M+1$ valuations $\{v^*_1, \ldots, v^*_{M+1}\}$.

Now, for each $i \in \{1, \ldots, M\}$ and $k \in \{1, \ldots, M+1\}$, let us assign boolean values to ${}^{(k)}\mathfrak{p}_{\nu^{-1}(i)}$ according to v^*_k:

$${}^{(k)}\mathfrak{p}_{\nu^{-1}(i)} \text{ is } \mathsf{true} \text{ iff } v^*_k \left(\mathfrak{p}_{\nu^{-1}(i)}\right) = 1.$$

The satisfiability of $(\mathsf{prop\_pos})$, $(\mathsf{prop\_prob})$ and $(\mathsf{prop\_phi})$ is an immediate consequence of the definition of v^*_k, for each $k \in \{1, \ldots M\}$.

For the remaining assertions, recall that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ is a model for δ. Hence, for each $j \in \{1, \ldots, m\}$, at least one of the following cases occur:

- either there exists $\ell \in \{1, \ldots, k_j\}$ such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \neg\forall\varphi^j_\ell$ and, in that case, exists $v_{\rho^{\varphi^j_\ell}} \in V_2$ such that $v_{\rho^{\varphi^j_\ell}}\left(\neg\mathsf{prop}_{\varphi^j_\ell}\right) = 1$ and so, for each $v^*_{\rho_\theta} \in V^*_1$, $v^*_{\rho_\theta}\left(\neg\mathsf{prop}^{[j,\ell]}_{\varphi^j_\ell}\right) = 1$, which implies that for each $k \in \{1, \ldots, M+1\}$, $v^*_k\left(\neg\mathsf{prop}^{[j,\ell]}_{\varphi^j_\ell}\right) = 1$ and so ${}^{(k)}\mathsf{prop}^{[j,\ell]}_{\neg\varphi^j_\ell}$ holds;

- or there exists $s \in \{1, \ldots, n_j\}$ such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \forall\psi^j_s$, so, for every $\rho \in \mathsf{S}$, $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho \Vdash_{\mathsf{loc}} \psi^j_s$. Since every valuation in $V_1 \cup V_2$ derives from an outcome in S, it follows that every valuation in $V_1 \cup V_2$ satisfies $\mathsf{prop}_{\psi^j_s}$ and by definition of $\alpha^*_{\psi^j_s}$ we have $\alpha^*_{\psi^j_s} = 1$;

- or there exists $r \in \{1, \ldots, s_j\}$ such that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P}) \Vdash \xi_r^j$, where ξ_r^j is a probabilistic literal of the form $q_{(r,j,1)} \cdot \mathsf{Pr}\left(\varphi_{(r,j,1)}\right) + \cdots + q_{(r,j,l_r^j)} \cdot \mathsf{Pr}\left(\varphi_{(r,j,l_r^j)}\right) \bowtie_r^j q_{(r,j)}$. Assume that $\nu\left(\varphi_{(r,j,s)}\right) = t_s$ for each $s \in \{1, \ldots, \ell_r^j\}$, and pay attention to the following sequence of (in)equalities:

$$\sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \alpha^*_{\varphi_{(r,j,s)}} = \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \alpha^*_{\nu^{-1}(t_s)} = \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \sum_{k=1}^{K} b^*_{t_s,k} = \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \left(\sum_{k=1}^{K} h^*_{t_s,k} \cdot \pi^*_k \right) =$$

$$= \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \left(\sum_{v^* \in V_1^*} v^*\left(\mathfrak{p}_{\nu^{-1}(t_s)}\right) \cdot \pi(v^*) \right) = \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \left(\sum_{\substack{\theta \in \Theta \\ \mathsf{S}^\theta \neq \varnothing}} v_{\rho_\theta}\left(\mathsf{prop}_{\nu^{-1}(t_s)}\right) \cdot \pi(v_{\rho_\theta}) \right) =$$

$$= \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \left(\sum_{\substack{\theta \in \Theta \text{ st } \mathsf{S}^\theta \neq \varnothing \\ \theta \to \varphi_{(r,j,s)}}} \pi(v_{\rho_\theta}) \right) = \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \left(\sum_{\substack{\theta \in \Theta \text{ st } \mathsf{S}^\theta \neq \varnothing \\ \theta \to \varphi_{(r,j,s)}}} \mu(\mathsf{S}^\theta) \right) =$$

$$= \sum_{s=1}^{\ell_r^j} q_{(r,j,s)} \mu(\mathsf{S}^{\varphi_{(r,j,s)}}) \bowtie_r^j q_{(r,j)} \,.$$

Hence we conclude that (prob) is satisfiable.

Additionally, each valuation in $V_1 \cup V_2$ satisfies Φ^δ since $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ satisfies all instances of Eq1-Eq4, DEq, E(Γ), D(Λ), and it follows that $\alpha^*_\phi = 1$ and so (prob_phi) is satisfied. We conclude that Algorithm 4.2 returns Sat.

For the reciprocal implication, assume that Algorithm 4.2 returns Sat. Let w denote the valuation over $\mathcal{B}^*$ that assigns Boolean values to the propositional symbols in $\mathcal{B}^*$ and satisfies the assertions (prop_pos), (prop_prob), (prop_phi), and let us mark as x^* the solution for each variable x in assertions (val1), (val2), (prob), (prob_phi), (sums1) and also (prob).

Below are presented several steps, where we refine the valuation w until reaching a final set of valuations $\mathcal{V}$, derived from w, defined over the set of propositional symbols $\mathcal{B}^\delta$.

First Step: Let $W = \{w_1, \ldots, w_{M+1}\}$ be the set of valuations such that, for each $k \in \{1, \ldots, M+1\}$, $w_k : \widetilde{\mathcal{B}} \to \{0, 1\}$ is defined from w as:

$$w_k(\mathsf{p}) = w\left({}^{(k)}\mathsf{p}\right).$$

Then, consider the probability distribution $\pi : W \to [0, 1]$ such that $\pi(w_k) = \pi^*_k$.

Second Step: Consider a subset $W_0 = \{w_{k_1}, \ldots, w_{k_z}\} \subseteq W$ of the valuations with positive probability: $W_0 = \{w \in W \mid \pi(w) > 0\}$. Due to assertion (sums1), $W_0 \neq \varnothing$.

It is time to highlight some remarks. Notice that for each $j \in \{1, \ldots, m\}$:

(i) either there exists $s \in \{1, \ldots, n_j\}$ such that $\alpha^*_{\psi_s^j} = 1$, which implies that $h^*_{\nu(\psi_s^j),k} = 1$ for every $k \in \{k_1, \ldots, k_z\}$, and so ${}^{(k)}\mathfrak{p}_{\psi_s^i}$ holds for every $k \in \{k_1, \ldots, k_z\}$ and, by (prop_pos), $w_k\left(\mathsf{prop}^{[j,\ell]}_{\psi_s^j}\right) = w_k\left(\mathsf{prop}_{\psi_s^j}\right) = 1$, for every $k \in \{k_1, ..., k_z\}$, $j \in \{1, ..., m\}$ and $\ell \in \{1, ..., k_j\}$;

(ii) or there exists $\ell \in \{1, \dots, k_j\}$ for which there exists $k_{[j,\ell]}$ such that ${}^{(k_{[j,\ell]})}\mathsf{prop}^{[j,\ell]}_{\neg\varphi^j_\ell}$ holds;

(iii) or either there exists $r \in \{1, \dots, s_j\}$ for which $\sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \cdot \alpha^*_{\varphi_{(r,j,s)}} \bowtie^j_r q_{(r,j)}$. Assume that, for each $s \in \{1, \dots, \ell^j_r\}$, $t_s = \nu(\varphi_{(r,j,s)})$, and let us ease notation and denote by $V^{\varphi} = \{w \in V \mid w\left(\mathfrak{p}_\varphi\right) = 1\}$. These leads to the following equivalences:

$$\begin{array}{rl}
& \sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \cdot \alpha^*_{\varphi_{(r,j,s)}} \bowtie^j_r q_{(r,j)} \\
\text{iff} & \sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \left(\sum\limits_{k=1}^{M+1} b^*_{t_s,k} \right) \bowtie^j_r q_{(r,j)} \\
\text{iff} & \sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \left(\sum\limits_{\substack{k \in \{k_1, \dots, k_z\} \\ h^*_{t_s,k} = 1}} \pi^*_k \right) \bowtie^j_r q_{(r,j)} \\
\text{iff} & \sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \left(\sum\limits_{w \in W_0^{\varphi_{(r,j,s)}}} \pi(w) \right) \bowtie^j_r q_{(r,j)} \\
\text{iff} & \sum\limits_{s=1}^{\ell^j_r} q_{(r,j,s)} \left(\sum\limits_{w \in W^{\varphi_{(r,j,s)}}} \pi(w) \right) \bowtie^j_r q_{(r,j)}.
\end{array}$$

Furthermore, since $\alpha^*_\phi = 1$, $v_k(\phi^{[j,\ell]}) = v_k(\phi) = 1$, for every $k \in \{k_1, \dots, k_z\}$, $j \in \{1, \dots, m\}$ and $\ell \in \{1, \dots, k_j\}$.

Third Step: Now consider a finest set of valuations over the set of propositional symbols $\mathcal{B}^\delta$:

$$\mathcal{V} = \bigcup_{j=1}^{m} \bigcup_{\ell=1}^{k_j} \left\{ v_{[j,\ell]} \mid {}^{(k_{[j,\ell]})}\mathsf{prop}^{[j,\ell]}_{\neg\varphi^j_\ell} \text{ holds} \right\} \cup \{v_{k_1}, \dots, v_{k_z}\} \subseteq \{0,1\}^{\mathcal{B}^\delta},$$

where, for each $\mathsf{p} \in \mathcal{B}^\delta$,

$$\begin{array}{rcl}
v_{[j,\ell]}(\mathsf{p}) & = & w_{k_{[j,\ell]}}\left(\mathsf{p}^{[j,\ell]}\right) \\
v_{k_1}(\mathsf{p}) & = & w_{k_1}(\mathsf{p}) \\
& \vdots & \\
v_{k_z}(\mathsf{p}) & = & w_{k_z}(\mathsf{p})
\end{array}$$

Let π_0 denote the probability distribution $\pi_0 : \mathcal{V} \to [0,1]$ defined by:

$$\begin{array}{rcll}
\pi_0(v_{k_1}) & = & \pi^*_{k_1} & \\
& \vdots & & \\
\pi_0(v_{k_z}) & = & \pi^*_{k_z} & \\
\pi_0(v) & = & 0 & \text{for any } v \in \mathcal{V} \setminus \{v_{k_1}, \dots, v_{k_z}\}
\end{array}$$

Notice that $\sum\limits_{v \in \mathcal{V}} \pi_0(v) = 1$.

Now we aim at deriving a model for δ form the valuations in $\mathcal{V}$. For this purpose, we follow the same reasoning as for $\mathsf{DNFSAT\text{-}DEqPrL}$ and extend F with new constants $c_{v,n}$, for each $n \in N$ and $v \in V$. Let F^* be an extension of F as described in (4.8). Then, consider the consistent set $M = \bigcup_{i \in I} M_i \subseteq \mathsf{Glob}^{\varnothing}_{\mathrm{F}^*}$ defined inductively on (4.9) and take a maximal consistent set Ξ^* extending M, whose existence is guaranteed by the Lindenbaum's Lemma. Consider the quotient F^*-algebra $\mathbb{A} = T_{\mathrm{F}^*}(\varnothing)_{/\equiv^*}$ derived from the congruence relation defined by $t_1 \equiv^* t_2$ iff $\forall(t_1 \approx t_2) \in \Xi^*$. The interpretations for domain names are taken according to the maximal consistent set Ξ^*, $\mathrm{I}^{\mathbb{A}} : \mathcal{D} \to \wp(A)$ is defined as $\mathrm{I}^{\mathbb{A}}(\mathrm{D}) = \{[t]_{\equiv} \mid \forall(t \in \mathrm{D}) \in \Xi^* \text{ and } t \in T_{\mathrm{F}^*}(\varnothing)\}$ for each $\mathrm{D} \in \mathcal{D}$.

Then, we translate valuations into outcomes, defining an outcome $\rho^v : N \to \mathbb{A}$ for each valuation $v \in V$ as $\rho^v(n) = [c_{v,n}]_{\equiv}$. The set of outcomes is the set $\mathsf{S} = \{\rho^v \mid v \in \mathcal{V}\}$. Note that $\mathsf{S} \neq \varnothing$ provided that $W_0 \neq \varnothing$. Regarding the probabilistic component, let us import the probabilities on the propositional level through the probability distribution $\mathcal{P} : \mathsf{S} \to [0,1]$ defined by $\mathcal{P}(\rho^v) = \pi_0(v)$ for each $\rho^v \in \mathsf{S}$. A probability space $\mathbb{P} = (\mathsf{S}, \mathscr{A}, \mu)$ is then defined using the discrete σ-algebra $\mathscr{A}$ over S, and the probability measure $\mu : \mathscr{A} \to [0,1]$ defined by $\mu(X) = \sum_{\rho \in X} \mathcal{P}(\rho)$.

The conclusion that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ is actually an F-structure follows by the straightforward verification that μ is a probability measure. The verification that $\mathbb{A}$ satisfies Γ, that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}})$ verifies Λ and that

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} \varphi \text{ iff } v\,(\mathsf{p}_\varphi) = 1 \text{ for each } \varphi \in \mathsf{Eq}(N) \cup \mathsf{DRes}(N):$$

results from the same observations used in the context of $\mathsf{DNFSAT\text{-}DEqPrL}$. Again, we conclude that for each $\varphi \in \mathsf{subform}(\delta) \cap \mathsf{Loc}$ and $v \in \mathcal{V}$,

$$(\mathbb{A}, \mathrm{I}^{\mathbb{A}}), \rho^v \Vdash_{\mathsf{loc}} \varphi \text{ iff } v(\mathsf{prop}_\varphi) = 1. \tag{4.17}$$

From (4.17), we conclude that $(\mathbb{A}, \mathrm{I}^{\mathbb{A}}, \mathbb{P})$ is model for δ just by recalling remarks (i)-(iii). □

The satisfiability proof gives us a way to effectively define the valuations that will lead to the final model of a satisfiable global formula. Let us illustrate this construction in a small example.

Example 4.4.11. Consider the signature $\mathrm{F}^{\mathrm{xor}}$, the equational theory Γ^{xor} and the axiomatization Λ^{xor} that we have been using and was introduced in Example 4.1.1. Furthermore, let us consider the satisfiable CNF global formula

$$\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \cdot \mathsf{Pr}(\neg n \in \mathsf{even}) \wedge \neg\forall(n \in \mathsf{even}) \wedge \neg\forall(n \in \mathsf{odd}).$$

Note that $\mathfrak{SLoc} = \{n \approx \mathsf{zero}, \neg n \in \mathsf{even}\}$ and consider $\nu(n \approx \mathsf{zero}) = 1$, $\nu(\neg n \in \mathsf{even}) = 2$, $\nu(\phi) = 3$.

Now, for instance, the assertion (prob) presented in page 114 would read like:

$$\left(\alpha_{n\approx\mathsf{zero}} \leq \frac{2}{3}\cdot\alpha_{\neg n\in\mathsf{even}}\right) \wedge (\bigvee_{k=1}^{3} {}^{(k)}\mathsf{prop}^{[2,1]}_{\neg n\in\mathsf{even}}) \wedge (\bigvee_{k=1}^{3} {}^{(k)}\mathsf{prop}^{[3,1]}_{\neg n\in\mathsf{odd}}).$$

Consider a solution for the $\mathsf{QF\_LIRA}$ problem composed by the values x^* for each variable x and by a valuation w over $\mathcal{B}^*$, and let us sketch the construction of the model.

Along the reciprocal implication we are driven among the several valuations that will afterwards lead to the outcomes that constitute the model for the given formula. We start by realizing the existence of a valuation $v_{k_{[2,1]}}$ such that $v_{k_{[2,1]}}(\mathsf{prop}^{[2,1]}_{\neg n\in\mathsf{even}}) = 1$ and another valuation $v_{k_{[3,1]}}$ such that $v_{k_{[3,1]}}(\mathsf{prop}^{[3,1]}_{\neg n\in\mathsf{odd}}) = 1$. Then, we are able to show up the final set of valuations, $v_{[2,1]}, v_{[3,1]}, v_1, v_2, v_3, v_4$, over $\mathcal{B}^\delta$, defined by:

$$\begin{aligned} v_{[2,1]}(\mathsf{p}) &= w_{k_{[2,1]}}(\mathsf{p}^{[2,1]}), \\ v_{[3,1]}(\mathsf{p}) &= w_{k_{[3,1]}}(\mathsf{p}^{[3,1]}), \\ v_k(\mathsf{p}) &= w_k(\mathsf{p}), \qquad \text{for each } \mathsf{p}\in\mathcal{B}^\delta,\ k\in\{1,2,3,4\}. \end{aligned}$$

The probability distribution π_0 is, in this way, strictly assigned to the last set of valuations:

$$\begin{aligned} \pi_0(v_{[2,1]}) &= 0, \\ \pi_0(v_{[3,1]}) &= 0, \\ \pi_0(v_k) &= \pi^*_k, \quad \text{for each } k. \end{aligned}$$

Note that at least one valuation satisfies each propositional formula corresponding to the existential conjuncts and the probability distribution is faithful to the probabilistic constraints. The model for the given global formula arises by a construction similar to the one described in the proof of completeness. △

Tseitin-like transformation on DEqPrL

So far, we have described two algorithms to decide the satisfiability of a global formulas. For $\mathsf{DNFSAT\text{-}DEqPrL}$, we argued that although we did not present the most efficient algorithm, transforming a global formula into DNF would not also be the best choice, as it eventually would lead to an explosion in the length of the formula. The same arguments can be used against the transformation of a global formula into CNF. Luckily, we also have a Tseitin-like transformation for DEqPrL, which provides us a method to transform any global formula into an equisatisfiable CNF formula with linear size on the length of the original formula, and allows us to take advantage of the $\mathsf{CNFSAT\text{-}DEqPrL}$ solver.

Following the same idea as for EqCL (see Section 2.4.1), the idea is to introduce additional atoms $\forall(n_1^{\delta'} \approx n_2^{\delta'})$ for every non-atomic subformula δ' of δ, ensure that $\forall(n_1^{\delta'} \approx n_2^{\delta'}) \leftrightarrow \delta'$ and, in the end, additionally ensure that the former formula is satisfied by imposing $\forall(n_1^{\delta} \approx n_2^{\delta})$.

In this sense, given a global formula $\delta \in \mathsf{Glob}$, we consider the set of all subformulas of δ that are not atoms, $\mathsf{subform}(\delta) \smallsetminus (\forall\mathsf{Loc} \cup \mathsf{Prob})$, and fix a pair of new (and distinct) names for each of them. To ease notation, we denote by $\mathsf{GA}(\delta')$ the atom corresponding to the subformula $\delta' \in (\mathsf{subform}(\delta) \smallsetminus (\forall\mathsf{Loc} \cup \mathsf{Prob}))$. Furthermore, we abuse notation and also denote an atom $\delta' \in (\mathsf{subform}(\delta) \cap (\forall\mathsf{Loc} \cup \mathsf{Prob}))$ by $\mathsf{GA}(\delta')$. In short,

$$\mathsf{GA}(\delta') = \begin{cases} \delta' & \text{if } \delta' \in (\forall\mathsf{Loc} \cup \mathsf{Prob}) \\ \forall(n_1^{\delta'} \approx n_2^{\delta'}) & \text{otherwise} \end{cases}$$

For each subformula $\delta' \in (\mathsf{subform}(\delta) \smallsetminus (\forall\mathsf{Loc} \cup \mathsf{Prob}))$, we define the additional conjuncts $\mathsf{tc}(\delta')$ representing the equivalence $\mathsf{GA}(\delta') \leftrightarrow \delta'$ in CNF according to the structure of δ':

$\mathsf{tc}(\neg\psi) = (\mathsf{GA}(\neg\psi) \vee \mathsf{GA}(\psi)) \wedge (\neg\mathsf{GA}(\neg\psi) \vee \neg\mathsf{GA}(\psi))$;

$\mathsf{tc}(\psi_1 \wedge \psi_2) = (\neg\mathsf{GA}(\psi_1 \wedge \psi_2) \vee \mathsf{GA}(\psi_1)) \wedge (\neg\mathsf{GA}(\psi_1 \wedge \psi_2) \vee \mathsf{GA}(\psi_2)) \wedge (\mathsf{GA}(\psi_1 \wedge \psi_2) \vee \neg\mathsf{GA}(\psi_1) \vee \neg\mathsf{GA}(\psi_2))$;

$\mathsf{tc}(\psi_1 \vee \psi_2) = (\mathsf{GA}(\psi_1 \vee \psi_2) \vee \neg\mathsf{GA}(\psi_1)) \wedge (\mathsf{GA}(\psi_1 \vee \psi_2) \vee \neg\mathsf{GA}(\psi_2)) \wedge (\neg\mathsf{GA}(\psi_1 \vee \psi_2) \vee \mathsf{GA}(\psi_1) \vee \mathsf{GA}(\psi_2))$.

We define the Tseitin-like transformation on DEqPrL simply as:

$$\mathsf{tt}(\delta) = \mathsf{GA}(\delta) \wedge \bigwedge_{\delta' \in (\mathsf{subform}(\delta) \smallsetminus (\forall\mathsf{Loc} \cup \mathsf{Prob}))} \mathsf{tc}(\delta').$$

Notice that the obtained CNF formula has linear size on the length of δ, since $\mathsf{subform}(\delta)$ has linear size on the length of δ and the transformation $\mathsf{tc}(\cdot)$ increments the length of the formula only by a constant. Furthermore, just like for EqCL (see Lemma 2.4.9), as a corollary of the previous construction we have the following Lemma.

Lemma 4.4.12. *Given $\delta \in \mathsf{Glob}$, there exists an equisatisfiable formula $\delta' \in \mathsf{Glob}$ in conjunctive normal form whose length is linear on the length of δ and can be computed in polynomial time.*

Example 4.4.13. In the context of Example 4.2.2, for instance, we can use the Tseitin-like transformation for DEqPrL to obtain an equisatisfiable formula in CNF for

$$(\forall(k \approx k^*) \vee \mathsf{Pr}(k \approx k^*) \geq \alpha) \rightarrow \mathsf{Pr}\left(\{\{n\}_k\}_{k^*}^{-1} \approx \pi_2(a,n)\right) \geq \alpha,$$

for some $0 \leq \alpha \leq 1$, as follows: begin by rewriting the formula without the connective →, introduced by abbreviation, and then identify its non-atomic subformulas:

$$\underbrace{\underbrace{\neg\overbrace{(\forall(k \approx k^*) \vee \mathsf{Pr}(k \approx k^*) \geq \alpha)}^{\delta_1}}_{\delta_2} \ \vee\ \mathsf{Pr}\left(\{\{n\}_k\}_{k^*}^{-1} \approx \pi_2(a,n)\right) \geq \alpha}_{\delta}.$$

The CNF formula equisatisfiable to δ is:

$$\mathsf{tt}(\delta) = \mathsf{GA}(\delta) \wedge \mathsf{tc}(\delta_1) \wedge \mathsf{tc}(\delta_2) \wedge \mathsf{tc}(\delta),$$

where

$$\mathsf{tc}(\delta_1) = (\mathsf{GA}(\delta_1) \vee \neg\forall(k \approx k^*)) \wedge (\mathsf{GA}(\delta_1) \vee \neg\mathsf{Pr}(k \approx k^*) \geq \alpha) \wedge (\neg\mathsf{GA}(\delta_1) \vee \forall(k \approx k^*) \vee \mathsf{Pr}(\{\{n\}_k\}^{-1}_{k^*} \approx \pi_2(a,n)) \geq \alpha),$$

$$\mathsf{tc}(\delta_2) = (\mathsf{GA}(\delta_2) \vee \mathsf{GA}(\delta_1)) \wedge (\neg\mathsf{GA}(\delta_2) \vee \neg\mathsf{GA}(\delta_1)),$$

$$\mathsf{tc}(\delta) = (\mathsf{GA}(\delta) \vee \neg\mathsf{GA}(\delta_2)) \wedge (\mathsf{GA}(\delta) \vee \neg\mathsf{Pr}\left(\{\{n\}_k\}^{-1}_{k^*} \approx \pi_2(a,n)\right) \geq \alpha) \wedge (\neg\mathsf{GA}(\delta) \vee \mathsf{Pr}(\{\{n\}_k\}^{-1}_{k^*} \approx \pi_2(a,n)) \geq \alpha).$$

△

SAT-DEqPrL **problem**

In general, we are looking for a procedure to decide the SAT-DEqPrL problem. Fortunately, the Tseitin-like transformation for DEqPrL and the CNFSAT-DEqPrL solver will greatly ease our task. Given a global formula $\delta \in \mathsf{Glob}$, we seek out an equisatisfiable formula δ' in CNF and then use the CNFSAT-DEqPrL solver to decide about the existence of a model for δ' (and for δ).

Theorem 4.4.14. *If Γ is a convergent equational theory and Λ is a set of domain clauses with the subterm property, then the* SAT-DEqPrL *problem is decidable.*

Proof. Given a global formula $\delta \in \mathsf{Glob}$, we use the Tseitin-like transformation for DEqPrL to convert δ into an equisatisfiable formula $\mathsf{tt}(\delta)$ in conjunctive normal form. Then, we run the CNFSAT-DEqPrL solver presented in Algorithm 4.2 on $\mathsf{tt}(\delta)$. If CNFSAT-DEqPrL returns Sat then $\mathsf{tt}(\delta)$ has a model, and so δ has a model; if it returns Unsat, then $\mathsf{tt}(\delta)$ is unsatisfiable and δ is also unsatisfiable. □

4.4.2 Validity

The decidability of the logic follows as an immediate corollary of the satisfiability result.

Theorem 4.4.15. *If Γ is a convergent equational theory and Λ is a set of domain clauses with the subterm property, then the logic is decidable.*

Proof. Since the deduction metatheorem holds, given a finite set $\Delta \subseteq \mathsf{Glob}$ and a formula $\varphi \in \mathsf{Glob}$, proving $\Delta \vdash_{(\Gamma,\Lambda)} \varphi$ is equivalent to proving that $\vdash_{(\Gamma,\Lambda)} ((\bigwedge_{\psi\in\Delta} \psi) \to \varphi)$, so we proceed by checking the decidability of the validity problem. Let $\delta \in \mathsf{Glob}$ be an arbitrary formula. We decide whether $\vdash_{(\Gamma,\Lambda)} \delta$ or $\nvdash_{(\Gamma,\Lambda)} \delta$ by testing the satisfiability of $\neg\delta$: if $\neg\delta$ is satisfiable, since the logic is sound, we conclude that $\nvdash_{(\Gamma,\Lambda)} \delta$; if $\neg\delta$ is not satisfiable, we can use completeness to conclude that $\vdash_{(\Gamma,\Lambda)} \delta$. □

Let us illustrate our decision procedure on a concrete example.

Example 4.4.16. Recall the signature $\mathrm{F}^{\mathrm{xor}}$, the equational theory Γ^{xor} and the axiomatization Λ^{xor}. Given $n \in \mathbb{N}$, we conclude that the formula

$$\left(\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \cdot \mathsf{Pr}(n \in \mathsf{even}) \wedge \forall(n \in \mathsf{even})\right) \rightarrow \left(\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \wedge \forall(\mathsf{suc}(n) \in \mathsf{odd})\right)$$

is valid provided that we proved in Example 4.4.9 that its negation is not satisfiable.

Similarly, we can use Example 4.4.11 to conclude that:

$$\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \cdot \mathsf{Pr}(\neg n \in \mathsf{even}) \nvdash_{(\Gamma^{\mathrm{xor}},\Lambda^{\mathrm{xor}})} \forall(n \in \mathsf{even}) \vee \forall(n \in \mathsf{odd}).$$

△

4.4.3 Complexity

The satisfiability result highlights a way of deciding SAT-DEqPrL by reduction to a QF_LIRA solver, under the assumption that Γ is a convergent equational theory and Λ is a set of domain clauses with the subterm property. In fact, our analysis revealed a reduction from SAT-DEqPrL to CNFSAT-DEqPrL and, furthermore, from CNFSAT-DEqPrL to QF_LIRA. Given the satisfiability result, we explored soundness and completeness of DEqPrL to derive the decidability result. We will now analyse complexity of the procedures previously obtained.

The complexity of CNFSAT-DEqPrL

As we already observed, the CNFSAT-DEqPrL solver presented in Algorithm 4.2 exhibits a way to transform a global formula δ written in CNF as $\bigwedge_{j=1}^{m}\left(\forall\psi_1^j \vee \ldots \vee \forall\psi_{n_j}^j \vee \neg\forall\varphi_1^j \vee \ldots \vee \neg\forall\varphi_{k_j}^j\right)$ into $\mathcal{O}(M + M \times (M+1))$ QF_LIRA assertions, where $M = \sum_{j=1}^{m}\left(n_j + k_j + \sum_{r=1}^{s_j} \ell_r^j\right) + 1$. Since Φ^δ has polynomial size on the length of δ, provided that Γ is given by means of a convergent rewriting system and Λ is a set of domain clauses with the subterm property, each assertion has polynomial size on the length of δ. For these reasons, Algorithm 4.2 exhibits a polynomial reduction from CNFSAT-DEqPrL to QF_LIRA.

The complexity result for the satisfiability problem CNFSAT-DEqPrL is parametric and also depends on the complexity of determining normal forms for terms with respect to the equational specification of the algebraic basis, which are fundamental to obtain the set of relevant terms $\mathsf{RelTerm}^\delta$. The complexity of CNFSAT-DEqPrL is the same as for QF_LIRA as long as the complexity of computing normal forms with respect to Γ (we dubbed it the *$\Gamma\downarrow$-problem*) is in P.

Corollary 4.4.17. *Assuming that Γ is a convergent equational theory whose $\Gamma\downarrow$-problem is in* P *and Λ is a set of domain clauses with the subterm property, then the satisfiability problem* CNFSAT-DEqPrL *is in* NP *and the validity problem for DNF formulas in* DEqPrL *is in* coNP.

Note that when the rewriting system underlying the equational theory Γ is subterm convergent, the complexity class of the $\Gamma\downarrow$-problem is in P. In fact, every term rewrites to a strict subterm in each rewriting step, so that, in the worst case, a term t takes $|\mathsf{subterms}(t)|$ steps until reaching its normal form, which is linear on the length of t. Since the unification algorithm also takes linear time (see [76, 92]), it follows that in this case the $\Gamma\downarrow$-problem is actually in P.

We should also remark that, as for EqCL, SAT can obviously be modeled in DEqPrL, by assigning an atom $\forall(t_1 \approx t_2)$ composed by two fresh terms t_1, t_2 to each propositional symbol to be considered.

Corollary 4.4.18. *If Γ is a subterm theory and Λ is a set of domain clauses with the subterm property, then* CNFSAT-DEqPrL *is* NP*-complete.*

The complexity of SAT-DEqPrL

The complexity result for SAT-DEqPrL follows immediately from the analysis of complexity of the CNFSAT-DEqPrL problem and from Lemma 4.4.12.

Corollary 4.4.19. *Assuming that Γ is a convergent equational theory whose $\Gamma\downarrow$-problem is in* P *and Λ is a set of domain clauses with the subterm property, then the satisfiability problem* SAT-DEqPrL *is in* NP *and the validity problem for* DEqPrL *is in* coNP*.*

Proof. The satisfiability procedure presented for DEqPrL reduces the analysis of the satisfiability of δ to the analysis of the satisfiability of its equisatisfiable formula $\mathsf{tt}(\delta)$ written in CNF. Lemma 4.4.12 allows us to conclude that $\mathsf{tt}(\delta)$ has linear length on the length of δ and can be computed in polynomial time. Consequently, we found out a linear reduction from SAT-DEqPrL to CNFSAT-DEqPrL. Provided that Γ is a convergent equational theory whose $\Gamma\downarrow$-problem is in P and Λ is a set of domain clauses with the subterm property, we conclude that SAT-DEqPrL is in NP. □

Corollary 4.4.20. *If the equational theory of Γ is generated by a subterm convergent rewriting system and Λ is a set of domain clauses with the subterm property, then the* SAT-DEqPrL *problem is* NP*-complete.*

4.4.4 Implementation

We implemented a DNFSAT-DEqPrL solver by reduction to QF_LIRA. The software was written in Python, and used Maude [37] for the rewriting reductions, and Yices [55] to solve the LIRA problem. The machine used for the tests was a Mac Pro at 3,33 GHz 6-Core Intel Xeon with 6 GB of memory. This work was developed with Carlos Caleiro and Filipe Casal. Our implementation is available in [28].

With a $\mathsf{DNFSAT\text{-}DEqPrL}$ solver in hands, we were able to test the examples that we have been proposing along this thesis. We now recall some examples that we have seen; they were tested with the implemented tool and their formulations can be found in [28].

- **Example 2.5.2:** Recall the characterization of the capabilities of the Dolev-Yao intruder where the cryptographic primitives satisfy the equational theory Γ^{DY}. We used the implemented $\mathsf{DNFSAT\text{-}DEqPrL}$ solver to verify the existence of an offline guessing attack to the cryptographic protocol presented in Example 2.5.2. For this purpose, we wanted to test whether:

  $$\forall(m_1 \approx (a, n_a) \wedge m_2 \approx \{n_a\}_{p_{ab}}) \nvdash_{\Gamma^{\mathsf{DY}}} \forall(\{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

  and

  $$\forall(m_1 \approx (a, n_a) \wedge m_2 \approx \{n_a\}_{p_{ab}}) \vdash_{\Gamma^{\mathsf{DY}}} \forall(p^*_{ab} \approx p_{ab} \to \{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

  We proved that, by testing the satisfiability of:

  $$\forall(m_1 \approx (a, n_a) \wedge m_2 \approx \{n_a\}_{p_{ab}}) \wedge \neg\forall(\{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

  and

  $$\forall(m_1 \approx (a, n_a) \wedge m_2 \approx \{n_a\}_{p_{ab}}) \wedge \neg\forall(p^*_{ab} \approx p_{ab} \to \{m_2\}^{-1}_{p^*_{ab}} \approx \pi_2(m_1)),$$

  respectively. As expected, the solver returns Sat to the former formula and Unsat for the latter. The running times were 51 seconds and 4 minutes, respectively.

- **Example 4.2.2:** Now, remind the setting where the Dolev-Yao intruder is extended with some cryptanalytic capabilities. In this context, the algebraic properties were axiomatized by Γ^{DY} and Λ^{DY}.

  We checked if whenever an attempt to guess the secret key k led to a message outside the scope of plaintexts, would mean that the value of k was not guessed correctly, i.e.,

  $$\forall(k \in \mathsf{sym\_key} \wedge m \in \mathsf{plaintext}) \vdash_{(\Gamma^{\mathsf{DY}}, \Lambda^{\mathsf{DY}})} \forall(\{\{m\}_k\}^{-1}_{k^*} \notin \mathsf{plaintext} \to k \not\approx k^*). \qquad (4.18)$$

  To decide that, we ran the implemented solver with the negation of the implication as input, i.e.,

  $$\forall(k \in \mathsf{sym\_key} \wedge m \in \mathsf{plaintext}) \wedge \neg\forall(\{\{m\}_k\}^{-1}_{k^*} \notin \mathsf{plaintext} \to k \not\approx k^*)$$

  and we concluded that it is actually unsatisfiable, so (4.18) holds.

  We also used the implemented $\mathsf{DNFSAT\text{-}DEqPrL}$ solver to conclude that even assuming that a guess k^* to the secret key k is indeed a symmetric key, guessing its concrete value is not simpler than decrypting a message encrypted with k. We tested the satisfiability of the following formula with $q = 0.4$:

  $$\mathsf{Pr}(k \approx k^*) = q \cdot \mathsf{Pr}(k^* \in \mathsf{sym\_key}) \wedge \forall(k^* \in \mathsf{sym\_key}) \wedge \neg\mathsf{Pr}(\{\{m\}_k\}^{-1}_{k^*} \approx m) \geq q.$$

  The running times were 4.7 seconds and 12 seconds, respectively.

- **Example 4.4.1:** Recall the algebraic characterization of the sum of single bits by the equational theory Γ^{xor} and with the domain restrictions Λ^{xor} introduced in Example 4.1.1. We concluded that, indeed, the global formula ξ presented in Example 4.4.16,

  $$\left(\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \cdot \mathsf{Pr}(n \in \mathsf{even}) \wedge \forall (n \in \mathsf{even})\right) \to \left(\mathsf{Pr}(n \approx \mathsf{zero}) \leq \frac{2}{3} \wedge \forall (\mathsf{suc}(n) \in \mathsf{odd})\right),$$

  is a theorem of DEqPrL by testing the satisfiability of $\neg\xi$ in the implemented software and attesting that it was actually unsatisfiable. The running times for each disjunct were 0.2 seconds and 0.5 seconds.

4.5 Applications to Information Security

Now we model some information security examples in DEqPrL and observe how important are the implementation details on the estimation of probabilities of the success of attacks to cryptographic protocols.

4.5.1 Offline Guessing Attacks with some Cryptanalysis

Let us recall the context of an offline guessing attack to a cryptographic protocol introduced in Definition 2.5.1. In a wider and more expressive formulation, we can model an attacker who, besides all the algebraic knowledge he has about the protocol and cryptographic primitives, is endowed with some cryptanalytic capabilities.

Recall that, to analyze offline guessing one assumes that an attacker has observed messages named $m_1, \ldots, m_k$ which were built as $t_1, \ldots, t_k \in T(N)$, but the attacker cannot know the concrete values of the random and secret names used to build them. Still, he can try to mount an attack by guessing some secrets $s_1, \ldots, s_n \in N$ used by the parties executing the protocol. The attack is successful if the attacker can distinguish whether his guesses $s_1^*, \ldots, s_n^*$ are correct or not. In DEqPrL, we can state a wider notion of offline guessing using cryptanalysis as described below.

Definition 4.5.1. Let $m_1, \ldots, m_k \in T(N)$ represent the messages exchanged by the parties executing a given cryptographic protocol, and Γ denote the equational specification of the underlying algebraic basis and Λ collects the domain restrictions on terms. The protocol is susceptible to an *offline guessing attack using cryptanalysis* if there exists a *recipe* $\varphi \in \mathsf{Loc}$, with $\mathsf{subterms}(\varphi) \subseteq T(\{m_1, \ldots, m_k, s_1^*, \ldots, s_n^*\})$ such that:

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \nvdash_{(\Gamma,\Lambda)} \quad \forall \varphi$$

and

$$\forall (m_1 \approx t_1 \wedge \cdots \wedge m_k \approx t_k) \quad \vdash_{(\Gamma,\Lambda)} \quad \forall (s_1^* \approx s_1 \wedge \cdots \wedge s_n^* \approx s_n \to \varphi)\,.$$

Although this definition seems very similar to Definition 2.5.1, note that now the recipe is a formula involving equations and domain restrictions and is constructed exclusively from messages observed by the attacker and from guesses for the secret values. The idea is the same: the recipe should not be derivable in general, but should be valid under the assumption that the attacker correctly guessed the secrets, proving to constitute a reliable formula for the attacker to check whether he actually guessed the secrets.

Again, this task is undecidable in general as the recipe may be arbitrarily complex, but recall that for subterm convergent rewriting systems the problem is decidable, as only a finite number of 'dangerous' recipes need to be tested [2,3,18].

The analysis of the existence of offline guessing attacks turns to be even more interesting when probabilities come into play, as the attacker will be able to narrow the set of possible secrets. In these lines, under appropriate probabilistic conditions and applying axiom $\mathsf{P3}$, one should be able to estimate the probability of offline guessing attacks within the logic.

Example 4.5.2. As an application, we can recall the protocol

$$1.\ a \to b : (a, n_a)$$

$$2.\ b \to a : \{n_a\}_{p_{ab}}$$

presented in Example 2.5.2 and which was proved to be attackable in EqCL (and so, in DEqPrL).

The existence of an offline guessing attack for this protocol led to an improvement of the exchanged messages by concatenating a confounder c with the nonce and encrypting with the public key $\mathsf{pk}(b)$ afterwards, giving rise to Gong's protocol [66]:

$$1.\ a \to b : \{\!|(n_a, c)|\!\}_{\mathsf{pk}(b)}$$

$$2.\ b \to a : \{n_a\}_{p_{ab}} \ .$$

Gong's protocol was proved to be secure against offline guessing [43,66], in the sense that the probability of an attack is negligible. We will observe that such security highly depends on the practical implementation of the protocol. This is one of the great achievements that we obtain with DEqPrL: we are able to cover some implementation details formally within the logic and conclude how do they compromise security.

Let us extend the set of domain names $\mathcal{D} = \mathcal{D}^{\mathsf{DY}} \cup \{\mathsf{conf}\}$ and, further, assume that the confounder c is sampled uniformly from a set with M elements, and that the set of symmetric keys from which p_{ab} is uniformly chosen has N elements. The estimation of the probability of an offline guessing attack on the independent names p_{ab} and c, with guesses p^*_{ab} and c^*, is given by:

$$\mathsf{Hyp} \vdash_{(\Gamma^{\mathsf{DY}},\Lambda^{\mathsf{DY}})} \mathsf{Pr}(p_{ab} \approx p^*_{ab} \wedge c \approx c^*) \leq \mathsf{Pr}(\{\!|(\{m_2\}^{-1}_{p^*_{ab}}, c^*)|\!\}_{\mathsf{pk}(b)} \approx m_1) \ ,$$

where the set of hypothesis consists of the uniform probabilities and independence of p^*_{ab} and c^*, of a record of the exchanged messages and of some cryptanalytic properties,

$$\mathsf{Hyp} = \{\forall(c^* \in \mathsf{conf}) \rightarrow \mathsf{Pr}(c \approx c^*) = \tfrac{1}{M},\ \ \forall(p^*_{ab} \in \mathsf{sym\_key}) \rightarrow \mathsf{Pr}(p_{ab} \approx p^*_{ab}) = \tfrac{1}{N},\ \ \mathsf{Ind}^{p^*_{ab},c^*}_{N,M}\ ,$$
$$\forall(c^* \in \mathsf{conf}),\ \ \forall(p^*_{ab} \in \mathsf{sym\_key}),\ \ \forall(m_1 \approx \{\!|(n_a, c_i)|\!\}_{\mathsf{pk}(b)}\ \ \wedge\ \ m_2 \approx \{n_a\}_{p_{ab}})\}\ .$$

According to the independence property for p^*_{ab} and c^*, the probability of guessing c and p_{ab}, and therefore the probability of success of an offline guessing attack is given by

$$\mathsf{Hyp} \vdash_{(\Gamma^{\mathsf{DY}},\Lambda^{\mathsf{DY}})} \frac{1}{N \cdot M}\ \ \leq\ \ \mathsf{Pr}(\{\!|(\{m_2\}^{-1}_{p^*_{ab}}, c^*)|\!\}_{\mathsf{pk}(b)} \approx m_1)\ \ .$$

Often, symmetric keys are defined as being *weak keys*, meaning that they are chosen from small sample spaces. In this sense, N is usually small. On the contrary, the commonly called *unguessable* values are believed to be chosen from very big sets. However, in the practical implementation of protocols it does not always happen, and we can model it in our logic. Notice that if M is also a small number, the probability of an attack is not negligible, as it is minimized by the non-negligible value $\frac{1}{N \cdot M}$.

△

4.5.2 On the Implementation Details

The reduced range of values taken by some critical parameters in the concrete implementation of cryptographic protocols can seriously compromise their security. Recently (see [10]) it was shown that some modern implementations of Diffie-Hellman key exchange are vulnerable to attacks from adversaries with reasonable resources.

A Diffie-Hellman key exchange consists of a preliminary agreement of a large prime p and a generator g by agents a and b, then both parties generate random integers x_a and x_b. Once all the values are fixed, a sends the exponentiation of g with x_a modulo p to b, and b sends the exponentiation of g with its private key x_b modulo p to a. At the end of the protocol, a and b are sharing the secret $(g^{x_a})^{x_b} \mathsf{mod}\ p\ \ =\ \ (g^{x_b})^{x_a} \mathsf{mod}\ p$.

1. $a \rightarrow b : g^{x_a}\ \mathsf{mod}\ p$
2. $b \rightarrow a : g^{x_b}\ \mathsf{mod}\ p$

Computing discrete logarithms remains the best known cryptanalytic attack to the security of Diffie-Hellman. In general, discrete log computations for arbitrary primes are known to take enough time to ensure that any session expires before the intruder carries out an attack, but Logjam [10] presents a technique that uses number field sieve and allows one to compute the discrete log of primes in a specified 512-bit group in about one minute, by means of a precomputation of the first three steps of number field sieve for that specific group. In fact, this vulnerability was already known since 1992 [22], but was applied by Logjam [10] to downgrade a TLS connection to use 512-bit Diffie-Hellman export-grade cryptography, through a man-in-the-middle network attacker. Let us analyze formally, within DEqPrL, how would a cryptanalytic attack through the discrete log compromise the security of Diffie-Hellman.

Example 4.5.3. Consider a Diffie-Hellman key exchange protocol:

1. $a \to b : g^{x_a} \ \mathsf{mod}\ p$
2. $b \to a : g^{x_b} \ \mathsf{mod}\ p$

Let us assume the attacker possesses enough computational resources to manage a precomputation of the first steps of number field sieve for a chosen group of 512-bit prime. Recall that the discrete logs in that group are then computed in a feasible amount of time. So, we can consider, in our signature, a function symbol representing the discrete log for each of those primes.

Consider the signature F^{DH} containing:

- $\mathsf{DLOG}_{(\cdot)}(\cdot,\cdot) \in \mathrm{F}^{\mathsf{DH}}_3$ representing an oracle for the discrete log of the subscript argument;
- $(\cdot)^{(\cdot)} \in \mathrm{F}^{\mathsf{DH}}_2$ representing exponentiation;
- $(\cdot)\ \mathsf{mod}\ (\cdot) \in \mathrm{F}^{\mathsf{DH}}_2$ representing the remainder of the division of the first by the second argument.

In the context of Diffie-Hellman key exchange, the equational properties of these operations are given by:

$$\Gamma^{\mathsf{DH}} = \{((x^{x_1})^{x_2}\ \mathsf{mod}\ x_3) \approx ((x^{x_2})^{x_1}\ \mathsf{mod}\ x_3)\}.$$

Now let us fix some domains, representing the chosen group of 512-bit primes for the implementation, the set of generators, the set of private keys and the set of ciphertexts:

$$\mathcal{D}^{\mathsf{DH}} = \{\ \mathsf{512\_prime},\ \ \mathsf{gen},\ \ \mathsf{priv\_key},\ \ \mathsf{ciphertext}\ \}.$$

We axiomatize the domain restrictions simply as:

$$\Lambda^{\mathsf{DH}} = \{\ \ (x \in \mathsf{priv\_key}, g \in \mathsf{gen}, p \in \mathsf{512\_prime} \Rightarrow (g^x\ \mathsf{mod}\ p) \in \mathsf{ciphertext})\ \ \}.$$

The probability of a cryptanalytic attack using discrete log can be expressed in DEqPrL as:

$$\mathsf{Hyp}^{\mathsf{DH}} \vdash_{(\Gamma^{\mathsf{DH}},\Lambda^{\mathsf{DH}})} \mathsf{Pr}(\mathsf{DLOG}_p(g,m_1) \approx x_a) \geq \mathsf{Pr}(p \in \mathsf{512\_prime})\ \ , \text{ where}$$

$$\mathsf{Hyp}^{\mathsf{DH}}\!=\!\left\{\forall(m_1\approx g^{x_a}\mathsf{mod}\ p\wedge m_2\approx g^{x_b}\mathsf{mod}\ p)\,,\forall(p\in 512\_\mathsf{prime}\!\rightarrow\!\mathsf{DLOG}_p(x_1,x_1^{x_2}\ \mathsf{mod}\ p)\approx x_2)\right\},$$

meaning that the probability of an offline guessing attack is bounded below by the probability of the intruder's smart choice for the group to which he develops the precomputation actually fall within the choice of the person who implemented the protocol.

Obviously, the attacker would not waste resources precomputing discrete logarithms unlikely to be used. There are groups of 512-bit primes known to be much popular than others, so the probability of the intruder's smart choice be within one of the implementer's choice can be significantly large, thereby influencing the probability of sucess of an attack.

This formalization should be seen as a simple illustration of how the cryptanalytic attacks can be modeled within DEqPrL. △

4.5.3 Privacy on e-voting

Recall the context of electronic voting protocols introduced in Subsection 2.5.2 and the always present privacy concern. Obviously, in Chapter 2 we underestimated privacy of e-voting protocols by strictly considering the case where the attacker was able to find out the user's vote. Easily, we realize that given a voting protocol we also would like to avoid cases where, even without guessing the concrete votes of users, the attacker could be able to conclude that two voters voted the same. We illustrate this case with an example.

Example 4.5.4. Recall Example 2.5.3, where we presented a very simple voting protocol in which users where asked to submit their votes, chosen among three possible values represented by $A, B, C \in F_0^{\mathsf{DY}}$, by sending an hash of their votes. Then, let $n \in \mathrm{F}_0^{\mathsf{DY}} \setminus \{A, B, C\}$ be a fixed constant and let us inspire in the work of Mödersheim et al. [82] to improve this voting protocol by masking user's vote. Recall the characterization of the cryptographic primitives by the set of Horn clauses Γ_h^{DY} extending Γ^{DY} with the collision-free property for the hash function $h \in \mathrm{F}_1^{\mathsf{DY}}$:

$$\Gamma_h^{\mathsf{DY}} = \Gamma^{\mathsf{DY}} \cup \{h(x) \approx h(y) \Rightarrow x \approx y\}.$$

Also, let us extend the set of domain clauses Λ^{DY} with domain restrictions for h and n:

$$\Lambda_h^{\mathsf{DY}} = \Lambda^{\mathsf{DY}} \cup \{(x \in \mathsf{plaintext} \Rightarrow h(x) \in \mathsf{ciphertext}), n \in \mathsf{plaintext}\}.$$

Consider the protocol where a voter a submits his vote v_a through an hash of the pairing (n, v_a) and a voter b submits his vote in the same way:

1. $a \rightarrow s : h((n, v_a))$
2. $b \rightarrow s : h((n, v_b))$

We easily observe that privacy is violated as long as the attacker can compare both messages and draw conclusions on whether a and b voted the same:

$$\forall(m_1 \approx h((n, v_a)) \wedge m_2 \approx h((n, v_b))) \vdash_{(\Gamma_h^{\mathsf{DY}}, \Lambda_h^{\mathsf{DY}})} \forall(m_1 \approx m_2 \leftrightarrow v_a \approx v_b).$$

Then, we conclude that in the presence of this voting protocol, where the mask n is the same for every voter, the intruder violates privacy with probability 1, as he is able to compare the votes from different users.

But we can take a step further on the estimation of the probability of an offline guessing attack, and conclude that the intruder has the possibility of effectively find out a user's vote. For this purpose, let us assume that:

- the probability of an user to vote A, B, C is $\alpha_A, \alpha_B, \alpha_C$, with $\alpha_A + \alpha_B + \alpha_C = 1$;
- the probability of guessing the constant n is α_n;
- the choice of the vote is independent of the choice of n.

Under these assumptions, we can effectively use DEqPrL to estimate the probability of an offline guessing attack as the product of the probability of guessing the vote v_a with the probability of guessing n:

$$\mathsf{Hyp} \vdash_{(\Gamma_h^{\mathsf{DY}}, \Lambda_h^{\mathsf{DY}})} \mathsf{Pr}(m_1 \approx h((n^*, v_a^*))) = \alpha_{v_a} \cdot \alpha_n \,,$$

where

$$\begin{aligned} \mathsf{Hyp} \;=\; \{ \;& \forall(m_1 \approx h((n, v_a)) \wedge m_2 \approx h((n, v_b)))\,, \mathsf{Pr}(v_a^* \approx v_a) = \alpha_{v_a}\,, \mathsf{Pr}(n^* \approx n) = \alpha_n\,, \\ & (\mathsf{Pr}(v_a^* \approx v_a) = \alpha_{v_a} \wedge \mathsf{Pr}(n^* \approx n) = \alpha_n) \rightarrow \mathsf{Pr}(v_a^* \approx v_a \wedge n^* \approx n) = \alpha_{v_a} \cdot \alpha_n \;\}. \end{aligned}$$

We could be wondering whether we could only achieve a lower bound for this probability, instead of the equality, by thinking that different components would lead to the same pairing. But note that it can never happen, provided the congruence property for the projections on the components.

We conclude that once the mask can be guessed, the secrecy of the vote is at risk. In the same lines, we could easily extend the analysis of the existence of offline guessing attacks to voting protocols where different and independent numbers n were chosen for each voter. △

4.6 Concluding Remarks

In a nutshell, we combined aspects from classical, equational, and probabilistic reasoning to construct a logic suited for the qualitative and quantitative analysis of equational constraints and domain restrictions over a set of outcomes. We began by defining a language that would allow us to make qualitative and quantitative assertions over local formulas. Once the semantics has been fixed, by endowing the set of possible outcomes with a probability space, we have also obtained a sound and weakly complete deductive system for DEqPrL, parametrized by a Horn-clause specification of the algebraic basis and by a set of domain clauses to characterize the domain restriction. Then, we paved the way to get a satisfiability result, by presenting a procedure for deciding DNFSAT-DEqPrL with calls to a GenPSAT oracle; we generalized the approach and found a polynomial reduction from CNFSAT-DEqPrL to QF_LIRA, provided that the algebraic basis is given by means of a convergent rewriting system and, additionally, that the axiomatization of domain restrictions enjoys a suitable subterm property. The complexity result followed naturally from the previous constructions. The decidability result also took advantage from the way in which the strategy was conducted, as it enabled the implementation of a prototype tool for DEqPrL using a QF_LIRA solver. This tool was tested in some simple examples and the running times were satisfactory on them, however there is still room for improvement with this implementation, namely in what concerns the interaction between the several softwares used. At the end of the chapter, we tested DEqPrL by verifying and estimating the probability of attacks to cryptographic protocols in the presence of an attacker with an informed way of cryptanalysis. In these examples we were able to model some implementation details that turned out to be very relevant for the analysis of security.

Again, even though our decidability results cover a very interesting range of examples, it would be interesting to explore their extension in order to handle decidable equational theories in general, i.e. not necessarily defined by means of convergent rewriting systems [51].

Chapter 5

Conclusions and Future Work

The main aim of this thesis was to provide a logic that would allow us to formalize the kind of reasoning inherent to the verification of security protocols, particularly in the context of offline guessing attacks. In order to take confidence on the interaction of the several components that came into play, we took a stepwise approach. We started by developing a logic involving equations, classical reasoning, and quantifiers (Chapter 2), then we pinpointed the need to explore a generalized probabilistic satisfiability problem (Chapter 3) and only then we were finally able to present the envisaged probabilistic logic over an algebraic basis with equations and domain restrictions (Chapter 4).

At the end of each chapter we have presented brief conclusions and remarks; where possible, we left open questions for future work, which arose as a result of a deeper understanding of the matter under consideration. Now we recall the main contributions of this thesis and point out some limitations, room for improvement and future research.

In Chapter 2 we presented EqCL - a logic that combines aspects from classical propositional logic, equational logic and quantifiers. We presented a sound and complete axiomatization parametrized by an equational specification of the algebraic basis. Then, we explored the satisfiability problem for EqCL and found a polynomial reduction to the SAT problem for classical propositional logic, under the assumption that the underlying equational theory was convergent. So far, this was the major milestone: we were able to automate the kind of reasoning used in the verification of, for instance, offline guessing attacks in a wide range of information security examples, like those that involve subterm convergent rewriting systems. In such (not that restraining) conditions, the satisfiability problem was proved to be NP-complete. This left open the opportunity to implement a prototype tool to decide this satisfiability problem by using off-the-shelf solvers, and techniques that are similar to those used in the SMT literature. Later, we had implemented an even wider prototype tool also involving a quantitative analysis by using Yices to solve the LIRA problems. We believe that an implementation of the satisfiability procedure for EqCL using a modern tool for solving SAT would lead to a more efficient performance, provided the very efficient way in which the SAT solvers are currently designed, albeit its NP-completeness.

In the context of the satisfiability problems, a generalization of the probabilistic satisfiability problem [59, 60] was instrumental for dealing with more expressive probabilistic expressions involving probabilistic formulas. In Chapter 3 we presented the GenPSAT problem and provided the theoretical framework that allowed the translation between GenPSAT and MIP problems. This enabled the implementation of a provably correct solver for GenPSAT. This translation was able to encode strict inequalities and disequalities into the MIP context. With the GenPSAT solver in hands, we detected and analyzed the phase transition behaviour. For the problems also covered by other solvers (e.g. SAT problems, PSAT problems) we cannot compete, in efficiency, with the existing solvers. We explored a reduction of GenPSAT to SMT and used both Yices and Z3 trying to achieve better running times, but it has not revealed to be more efficient in general. Imposing assertions (val2) strictly on the propositional variables used to define probabilistic restrictions, instead of doing it in every single fixed propositional variable, seems to enhance the performance in some cases. Of course, the importance of the choice of the programming language and the way it interacts with the MIP solver cannot be swept aside. That said, our prototype tool should be valued by its expressiveness and not by its efficiency on existing problems with off-the-shelf optimized solvers.

Once we came back to the equational context, in Chapter 4, DEqPrL has arisen naturally. DEqPrL is a probabilistic logic over an algebraic basis, including equations and domain restrictions, that combines aspects from classical propositional logic and equational logic with an exogenous approach to quantitative probabilistic reasoning. We presented a sound and (weak) complete deductive system, parametrized by an equational specification of the algebraic basis coupled with the intended domain restrictions. Then, we came up with a satisfiability procedure, under the assumption that the equational basis was given by means of a convergent rewriting system and, additionally, that the axiomatization of domain restrictions enjoyed a suitable subterm property. Such satisfiability procedure was developed through a polynomial reduction to QF_LIRA, inspired by the developments with GenPSAT. An implementation of the satisfiability problem took shape and was developed using Maude for the rewriting reductions and Yices to solve the LIRA problem. It was tested on the examples we had put forward. There is still room for improvement with this implementation, for many reasons: we assumed that the input was given in DNF, which suffer the limitations that we already examined; the software calls Maude and Yices many times, which slows down the running time. For lack of time we did not have the opportunity to implement the CNFSAT-DEqPrL procedure described in Algorithm 4.2, but would be interesting to do it. Finding a polynomial reduction to SAT would also be very useful for performance purposes.

The logic DEqPrL was applied to meaningful examples in information security and was proven to be very effective in incorporating some implementation details formally and in drawing conclusions about their impact in security, namely by verifying and estimating the probability of sucess of attacks to some cryptographic protocols. The application of the logic to many more information security examples would be very interesting! The combination of the procedure for the satisfiability problem for DEqPrL with the results in [2,3,18] that come up with the 'dangerous' recipes to test offline guessing attacks would be an asset to automatically decide the existence of offline guessing attacks or estimate its probability in cryptographic protocols whose cryptographic primitives are given by subterm convergent rewriting systems. Even so, a general tool like this will never be as effective as dedicated techniques [19,36,39,40].

Bibliography

[1] M. Abadi and B. Blanchet. Analyzing security protocols with secrecy types and logic programs. *Journal of the ACM*, 52(1):102–146, 2005.

[2] M. Abadi and V. Cortier. Deciding knowledge in security protocols under (many more) equational theories. In *Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 62–76. IEEE, 2005.

[3] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 367(1):2–32, 2006.

[4] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 36–47. ACM, 1997.

[5] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[6] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Proceedings of the 10th European Symposium on Research in Computer Security*, volume 3679 of *Lecture Notes in Computer Science*, pages 374–396. Springer, 2005.

[7] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness and completeness of formal encryption: The cases of key cycles and partial information leakage. *Journal of Computer Security*, 17(5):737–797, 2009.

[8] P. Adão, P. Mateus, T. Reis, and L. Viganò. Towards a quantitative analysis of security protocols. *Electronic Notes in Theoretical Computer Science*, 164(3):3–25, 2006.

[9] P. Adão, P. Mateus, and L. Viganò. Protocol insecurity with a finite number of sessions and a cost-sensitive guessing intruder is NP-complete. *Theoretical Computer Science*, 538:2–15, 2014.

[10] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 5–17. ACM, 2015.

[11] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P. C. Héam, O. Kouchnarenko, J. Mantovani, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *International Conference on Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer, 2005.

[12] S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[13] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, 1999.

[14] M. Backes, D. Hofheinz, and D. Unruh. CoSP: A general framework for computational soundness proofs. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pages 66–78. ACM, 2009.

[15] M. Backes, C. Hritcu, and M. Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *Proceedings of the 21st Computer Security Foundations Symposium*, pages 195–209. IEEE, 2008.

[16] G. Bana, P. Adão, and H. Sakurada. Computationally complete symbolic attacker in action. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 18. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[17] D. Basin, M. D'Agostino, D. M. Gabbay, S. Matthews, and L. Viganò. *Labelled deduction.* Springer Science & Business Media, 2012.

[18] M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 16–25. ACM, 2005.

[19] M. Baudet, V. Cortier, and S. Delaune. YAPA: A generic tool for computing intruder knowledge. In *International Conference on Rewriting Techniques and Applications*, pages 148–163. Springer, 2009.

[20] B. Becker, C. Dax, J. Eisinger, and F. Klaedtke. LIRA: Handling constraints of linear arithmetics over the integers and the reals. In *International Conference on Computer*

Aided Verification, volume 4590 of *Lecture Notes in Computer Science*, pages 307–310. Springer, 2007.

[21] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 394–403. IEEE, 1997.

[22] B. Beurdouche, K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, A. Pironti, P. Y. Strub, and J. K. Zinzindohoue. A messy state of the union: Taming the composite state machines of tls. In *Symposium on Security and Privacy 2015*, pages 535–552. IEEE, 2015.

[23] A. Biere, M. Heule, and H. van Maaren. *Handbook of Satisfiability*, volume 185. IOS Press, 2009.

[24] G. Birkhoff. On the structure of abstract algebras. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 433–454. Cambridge University Press, 1935.

[25] B. Blanchet. A computationally sound mechanized prover for security protocols. *IEEE Transactions on Dependable and Secure Computing*, 5(4):193–207, 2008.

[26] G. Boole. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. Dover Publications, 1854.

[27] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 426, pages 233–271. The Royal Society, 1989.

[28] C. Caleiro, F. Casal, and A. Mordido. DNFSAT-DEqPrL solver, 2016. Available online at https://github.com/fcasal/satdeqprl.git.

[29] C. Caleiro, F. Casal, and A. Mordido. Generalized probabilistic satisfiability. SQIG - Instituto de Telecomunicações and IST - U Lisboa, Portugal, 2016. Submitted for publication. Available online at http://sqig.math.ist.utl.pt/pub/CaleiroC/16-CCM-genpsat.pdf.

[30] C. Caleiro, F. Casal, and A. Mordido. GenPSAT solver, 2016. Available online at https://github.com/fcasal/genpsat.git.

[31] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[32] V. Chandru and J. Hooker. *Optimization methods for logical inference*, volume 34. John Wiley & Sons, 2011.

[33] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *Proceedings of IJCAI 1991*, pages 331–337, 1991.

[34] A. Church. *Introduction to mathematical logic*, volume 13. Princeton University Press, 1996.

[35] V. Chvátal. *Linear programming.* Macmillan, 1983.

[36] Ş. Ciobâcă, S. Delaune, and S. Kremer. Computing knowledge in security protocols under convergent equational theories. In *International Conference on Automated Deduction*, pages 355–370. Springer, 2009.

[37] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martı-Oliet, J. Meseguer, and J. F. Quesada. Maude: specification and programming in rewriting logic. *Theoretical Computer Science*, 285(2):187–243, 2002.

[38] P. M. Cohn. *Universal algebra*, volume 6. Springer Science & Business Media, 2012.

[39] B. Conchinha, D. Basin, and C. Caleiro. Efficient decision procedures for message deducibility and static equivalence. In *Proceedings of 7th International Workshop on Formal Aspects in Security and Trust*, volume 6561 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2010.

[40] B. Conchinha, D. Basin, and C. Caleiro. FAST: An efficient decision procedure for deduction and static equivalence. In *Proceedings of 22nd International Conference on Rewriting Techniques and Applications*, volume 10 of *LIPIcs*, pages 11–20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010.

[41] B. Conchinha, D. Basin, and C. Caleiro. Symbolic probabilistic analysis of off-line guessing. In *European Symposium on Research in Computer Security*, volume 8134 of *Lecture Notes in Computer Science*, pages 363–380. Springer, 2013.

[42] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971.

[43] R. J. Corin and S. Etalle. A simple procedure for finding guessing attacks. 2004.

[44] V. Cortier, S. Kremer, and B. Warinschi. A survey of symbolic methods in computational analysis of cryptographic systems. *Journal of Automated Reasoning*, 46(3-4):225–259, 2011.

[45] F. G. Cozman and L. F. di Ianni. Probabilistic satisfiability and coherence checking through integer programming. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 7958 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2013.

[46] H. B. Curry and R. Feys. Combinatory Logic, volume I of Studies in Logic and the Foundations of Mathematics, 1958.

[47] A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In *Proceedings of the 32nd International Colloquium on Automata, Languages, and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2005.

[48] G. De Bona, F. G. Cozman, and M. Finger. Generalized probabilistic satisfiability through integer programming. *Journal of the Brazilian Computer Society*, 21(1):1–14, 2015.

[49] L. De Moura and N. Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.

[50] S. Delaune, S. Kremer, and M. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.

[51] N. Dershowitz, M. Okada, and G. Sivakumar. Canonical conditional rewrite systems. In *Proceedings of the 9th International Conference on Automated Deduction*, volume 310 of *Lecture Notes in Computer Science*, pages 538–549. Springer, 1988.

[52] W. Diffie and M. Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[53] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[54] P. H. Drielsma, S. Mödersheim, and L. Vigano. A formalization of off-line guessing for security protocol analysis. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 363–379. Springer, 2005.

[55] B. Dutertre and L. De Moura. The Yices SMT Solver. *Tool paper at http://yices.csl.sri.com/tool-paper.pdf*, 2006.

[56] H. Ehrig and B. Mahr. Fundamentals of algebraic specification 1. *EATCS Monograph in Theoretical Computer Science.*

[57] T. Fabrega, F. Javier, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(2-3):191–230, 1999.

[58] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1):78–128, 1990.

[59] M. Finger and G. De Bona. Probabilistic satisfiability: Logic-based algorithms and phase transition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 528–533, 2011.

[60] M. Finger and G. De Bona. Probabilistic satisfiability: algorithms with the presence and absence of a phase transition. *Annals of Mathematics and Artificial Intelligence*, 75(3-4):351–389, 2015.

[61] H. Ganzinger and R. Nieuwenhuis. Constraints and theorem proving. In *Constraints in Computational Logics: theory and applications*, volume 2002 of *Lecture Notes in Computer Science*, pages 159–201. Springer, 2001.

[62] I. P. Gent and T. Walsh. The hardest random SAT problems. In *KI-94: Advances in Artificial Intelligence*, pages 355–366. Springer-Verlag, 1994.

[63] G. Gentzen. Investigations into logical deduction. *American philosophical quarterly*, 1(4):288–306, 1964.

[64] G. Georgakopoulos, D. Kavvadias, and C. H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4(1):1–11, 1988.

[65] C. P. Gomes, H. Kautz, A. Sabharwal, and B. Selman. Satisfiability solvers. *Foundations of Artificial Intelligence*, 3:89–134, 2008.

[66] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Selected Areas in Communications*, 11(5):648–656, 1993.

[67] G. A. Gratzer. *Universal algebra*. Springer Science & Business Media, 2008.

[68] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[69] L. Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, 14(03):159–166, 1949.

[70] R. Impagliazzo and B. M. Kapron. Logics for reasoning about cryptographic constructions. In *Proceedings of the 44th Symposium on Foundations of Computer Science*, pages 372–383. IEEE, 2003.

[71] C. Kirchner and H. Kirchner. Equational logic and rewriting. *Logic and Computation*, 9:255–282, 2014.

[72] R. Kowalski. *Logic for problem solving*, volume 7. Ediciones Díaz de Santos, 1979.

[73] S. Kremer and M. Ryan. Analysis of an electronic voting protocol in the applied pi calculus. In *European Symposium on Programming*, pages 186–200. Springer, 2005.

[74] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, 1996.

[75] G. Lowe. A hierarchy of authentication specifications. In *Proceedings of the 10th Computer Security Foundations Workshop*, pages 31–43. IEEE, 1997.

[76] A. Martelli and U. Montanari. *Unification in linear time and space: A structured presentation.* Internal Report B76-16, Istituto di Elaborazione della Informazione, Consiglio Nazionale delle Ricerche, 1976.

[77] 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.

[78] G. F. McNulty. A field guide to equational logic. *Journal of Symbolic Computation*, 14(4):371–397, 1992.

[79] C. Meadows. Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21(1):44–54, 2003.

[80] K. Meinke and J. V. Tucker. *Universal algebra.* University of Wales (Swansea). Mathematics and Computer Science Division, 1991.

[81] E. Mendelson. *Introduction to mathematical logic.* CRC press, 2009.

[82] S. A. Mödersheim, T. Groß, and L. Viganò. Defining privacy is supposed to be easy. In *Proceedings of the 19th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, volume 8312 of *Lecture Notes in Computer Science*, pages 619–635. Springer, 2013.

[83] B. Montalto and C. Caleiro. Modeling and reasoning about an attacker with cryptanalytical capabilities. *Electronic Notes in Theoretical Computer Science*, 253(3):143–165, 2009.

[84] A. Mordido and C. Caleiro. An equation-based classical logic. In *Proceedings of the 22nd International Workshop on Logic, Language, Information, and Computation*, volume 9160 of *Lecture Notes in Computer Science*, pages 38–52. Springer, 2015.

[85] A. Mordido and C. Caleiro. Probabilistic logic over equations and domain restrictions. SQIG - Instituto de Telecomunicações and IST - U Lisboa, 2015. Submitted for publication. Available online at http://sqig.math.ist.utl.pt/pub/CaleiroC/15-MC-probeq.pdf.

[86] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

[87] R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Solving SAT and SAT Modulo Theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL (T). *Journal of the ACM*, 53(6):937–977, 2006.

[88] N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[89] P. Padawitz. *Computing in Horn clause theories*, volume 16. Springer Science & Business Media, 2012.

[90] C. H. Papadimitriou. *Computational complexity.* John Wiley and Sons Ltd., 2003.

[91] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity.* Courier Corporation, 1982.

[92] M. S. Paterson and M. N. Wegman. Linear unification. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pages 181–186. ACM, 1976.

[93] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1, 2):85–128, 1998.

[94] D. Pavlovic and C. Meadows. Bayesian authentication: Quantifying security of the Hancke- -Kuhn protocol. *Electronic Notes in Theoretical Computer Science*, 265:97–122, 2010.

[95] G. Peano. *Arithmetices principia: nova methodo.* Fratres Bocca, 1889.

[96] J. Pearl. *Do We Need Higher-order Probabilities And, If So, what Do They Mean?* UCLA, Computer Science Department, 1987.

[97] D. Pigozzi. Equational logic and equational theories of algebras. 1975.

[98] E. L. Post. Introduction to a general theory of elementary propositions. *American Journal of Mathematics*, 43(3):163–185, 1921.

[99] I. Ray and N. Narasimhamurthi. An anonymous electronic voting protocol for voting over the internet. In *Proceedings of the 3rd Internation Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 188–190. IEEE, 2001.

[100] J. Rosenhouse. *The Monty Hall problem: the remarkable story of Math's most contentious brain teaser.* Oxford University Press, 2009.

[101] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th annual ACM Symposium on Theory of Computing*, pages 216–226. ACM, 1978.

[102] A. Sernadas and C. Sernadas. *Foundations of logic and theory of computation.* College Publications, 2008.

[103] J. R. Shoenfield. *Mathematical logic*, volume 21. Addison-Wesley Reading, 1967.

[104] D. R. Stinson. *Cryptography: theory and practice.* CRC press, 2005.

[105] A. Tarski. Equational logic and equational theories of algebras. *Studies in Logic and the Foundations of Mathematics*, 50:275–288, 1968.

[106] A. Tarski. *Logic, semantics, metamathematics: papers from 1923 to 1938.* Hackett Publishing, 1983.

[107] G. S. Tseitin. On the complexity of derivation in propositional calculus. In *Automation of Reasoning*, pages 466–483. Springer, 1983.

[108] J. Van Eijck and F. Schwarzentruber. Epistemic probability logic simplified. *Advances in Modal Logic*, 10:158–177, 2014.

[109] L. Viganò. *Labelled non-classical logics.* Springer Science & Business Media, 2013.

[110] R. Wójcicki. *Theory of logical calculi: basic theory of consequence operations*, volume 199. Springer Science & Business Media, 2013.