UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Combining Satisfiability Procedures and Probabilistic Satisfiability

Filipe Manuel Rodrigues Casal

Supervisor: Doctor João Filipe Quintas dos Santos Rasga

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

Jury final classification: Pass with Distinction

2018

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Combining Satisfiability Procedures and Probabilistic Satisfiability

Filipe Manuel Rodrigues Casal

Supervisor: Doctor João Filipe Quintas dos Santos Rasga

Thesis approved in public session to obtain the PhD Degree in

Information Security

Jury final classification: Pass with Distinction

Jury

Chairperson: Doctor António Manuel Pacheco Pires, Instituto Superior Técnico da Universidade de Lisboa

Members of Committee:

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

Doctor Manuel António Gonçalves Martins, Universidade de Aveiro

Doctor Jaime Arsénio de Brito Ramos, Instituto Superior Técnico da Universidade de Lisboa

Doctor João Filipe Quintas dos Santos Rasga, Instituto Superior Técnico da Universidade de Lisboa

Funding Institutions

Fundação para a Ciência e Tecnologia

2018

Resumo

Nesta tese, propomo-nos estudar duas áreas essenciais para o estudo formal de sistemas complexos. Muitas vezes, sistemas complexos são compostos por várias componentes independentes e, como tal, para aplicarmos métodos formais nestes sistemas, tais como procedimentos de decisão da satisfação de fórmulas, é necessário compreender como combinar estes métodos de maneira modular. Por outro lado, estes sistemas complexos não são perfeitos, as suas componentes podem ser instáveis, ou os agentes que interagem com o sistema podem não ser determinísticos. Assim, para expressarmos muitas das propriedades interessantes a estudar nestes sistemas, necessitamos de uma lógica capaz de raciocinar probabilisticamente.

Começamos, no estudo de combinação de procedimentos de satisfação, por adaptar o conceito de teoria *shiny* para lógicas *many-sorted*. Com isso, fechamos a questão da equivalência de teorias *shiny* e *strongly polite* ao mostrar que, para teorias com um procedimento de decisão da satisfação para fórmulas sem quantificadores decidível, a classe de teorias *many-sorted shiny* coincide com a classe de teorias *many-sorted strongly polite*. Capitalizando nesta equivalência, obtemos também um método de combinação Nelson-Oppen para teorias *many-sorted shiny*.

Depois, estudamos um problema de satisfação probabilístico generalizado, GenPSAT, que consiste em decidir a satisfação de sistemas de desigualdades que envolvem probabilidades de fórmulas da lógica clássica proposicional. Em seguida, apresentamos uma redução polinomial a Programação Linear Inteira Mista que é utilizada para implementar uma ferramenta que decide GenPSAT. Esta ferramenta permite-nos depois estudar o comportamento de transição de fase deste problema NP-completo. Em seguida, apresentamos uma extensão natural de GenPSAT que permite combinações Booleanas das suas fórmulas. Esta extensão, GGenPSAT, tem uma linguagem que coincide precisamente com a linguagem da lógica probabilística de Fagin et al. Similarmente a GenPSAT, com o intuito da implementação de uma ferramenta que decida o problema da satisfação, desenvolvemos uma redução polinomial, desta vez a Satisfação Módulo Teorias, na teoria sem quantificadores da aritmética linear de inteiros e reais. É mais uma vez efectuado o estudo do fenómeno de transição de fase e apresentamos exemplos de aplicação da ferramenta no contexto de verificação de *hardware* e ataques de *side-channel*.

Por fim, este último exemplo de aplicações a ataques de *side-channel* é desenvolvido, onde formalizamos probabilisticamente a noção de *perfect-masking* na presença de atacantes capazes de realizar ataques de *side-channel*. Além da modelação de várias classes de atacantes, provamos que decidir se uma fórmula é *perfectly masked* na presença de atacantes poderosos está em co-NP.

Palavras-chave: Combinação de Procedimentos de Satisfação, Método de Nelson-Oppen, Satisfação Probabilística, Transição de Fase, Procedimentos de Satisfação Probabilística, Ataques de Side-channel

Abstract

In this thesis, we aim to study two areas that are essential to the application of formal methods to real-world systems. The highly complex systems that are real-world systems are composed of several different and sometimes independent components. As such, solving satisfiability problems in these systems means that we need to know how to modularly combine satisfiability solvers for different logics. On the other hand, real-world systems are not flawless, their components are unreliable or they are embedded in a world where the agents that interact with the system are not deterministic in nature. As such, these systems need to be specified in logics with the ability to express probabilistic assertions.

Focused on the problem of combining satisfiability procedures, we begin by adapting the concept of shiny theory to the many-sorted case. With this notion in place, we close the question of the equivalence of shiny and strongly polite theories by establishing that, for theories with a decidable quantifier-free satisfiability problem, the set of many-sorted shiny theories coincides with the set of many-sorted strongly polite theories. Capitalizing on this equivalence, we obtain a Nelson-Oppen combination theorem for many-sorted shiny theories.

We then analyse 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. Furthermore, we implement and test a solver for the GenPSAT problem and as previously observed for many other NP-complete problems, we are able to detect a phase transition behaviour. Afterwards, we naturally generalize GenPSAT to allow Boolean combinations of linear inequalities involving probabilities of classical propositional formulas. GGenPSAT coincides precisely with the satisfiability problem of the probabilistic logic of Fagin et al. and was proved to be NP-complete. Here, we present a polynomial reduction of GGenPSAT to SMT over the quantifier-free theory of linear integer and real arithmetic. Capitalizing on this translation, we implement and test a solver for the GGenPSAT problem, study its phase transition behaviour and exemplify how to use this formalism to model two problems in information security, namely in hardware verification and side-channel attacks.

We conclude with applications to information security, namely side-channel attacks, and provide a probabilistic formalization of perfect masking when systems are under attack by agents with side-channel capabilities. Besides modelling several classes of attackers, we also show that deciding whether a formula is perfectly masked under such attackers is in co-NP.

Keywords: Combination of Satisfiability Procedures, Nelson-Oppen Method, Probabilistic Satisfiability, Phase Transition, Probabilistic Satisfiability Solver, Side-channel Attacks

Acknowledgments

I would like to thank my supervisor Prof. João Rasga for all the attention and support provided in the last few years.

Then, I would like to thank my co-authors with whom I learned so much with our discussions: Andreia Mordido, Carlos Caleiro, João Rasga and André Souto.

A special thanks also goes to all the members of STT, especially jofra, mrsilva and jcfg, as well as our mentor Prof. Pedro Adão.

Finally I would like to thank all the PMI brothers and sisters, office mates and friends for the wonderful time shared during these last years.

A sincere obrigado also goes to my family for the unconditional support they have always given me.

This work has been done under the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER–PT2020) and partially supported by Fundação para a Ciência e a Tecnologia by way of grant UID/MAT/04561/2013 to Centro de Matemática, Aplicações Fundamentais e Investigação Operacional of Universidade de Lisboa (CMAF-CIO). Furthermore, I also acknowledge the support from the DP-PMI and FCT (Portugal) through scholarship SRFH/BD/52243/2013.

Contents

Introduction

The impact that the fields of logic and related formal methods have in other areas of thought is undeniable. This impact ranges from the verification of safety properties in real-life systems [CW96, SSS00], hardware [Gup92, MS95] and software [DKW08, KEH+09] bug finding, proving correctness of software specifications to proving correctness of cryptographic protocols [Mea94, AG97, CHSvdM16], and detecting the existence of side-channel attacks in software specifications [EWS14]. Also, Artificial Intelligence as a whole has seen successful applications of logic and formal methods [GHR98], either to solve hard problems such as planning [BK00] or to formalize the interaction of agents and their knowledge through epistemic logics [WJ94, WJ95].

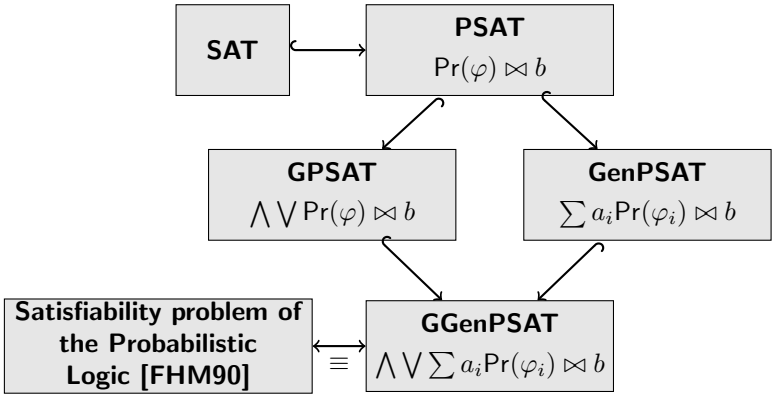
To solve these problems, we require a formal framework in which we specify our problem. In these *logics*, which are potentially very distinct from each other, one is able to define two problems: the *satisfiability problem* [Coo71, BHvM09] and the *model checking problem* [CGP99]. The satisfiability problem consists in, deciding if a certain fact is possible under the logical system we are considering. In case it is, we are provided with a model, or world, where this actually happens. On the other hand, in the model checking problem we are given a model of some logic and a formal specification, and our goal is to decide whether this model complies with such new set of rules. These problems are very distinct in nature and we will only focus on satisfiability problems in this work.

The highly complex systems that are real-world systems are composed of several different and sometimes independent components. As such, solving satisfiability problems in these systems means that we need to know how to combine satisfiability solvers for different logics. This problem has been extensively studied [NO79, Opp80, TH96, RRZ05, TZ05, JB10a, CR13] and these techniques, as well as others not studied here, are of the utmost importance as most modern general purpose SMT solvers such as Z3 [DMB08] and Yices [Dut14] uses them to combine the different theory solvers they are able to reason over. In the first chapter of this work, we focus on this area of formal methods that deals with combination of satisfiability procedures. In particular, we compare classes of theories whose satisfiability procedures are able to be combined with the satisfiability procedure of an arbitrary theory. In fact, we begin by extending the notion of shiny theory to the many-sorted case and then proceed to prove that the class of many-sorted shiny theories coincides with the class of many-sorted strongly polite theories. With this, we are able to propose a method to combine the satisfiability procedure for a many-sorted shiny theory with an arbitrary theory.

On the other hand, real-world systems are not flawless, their components are unreliable or they are embedded in a world where the agents that interact with the system are not deterministic in nature. As such, these systems need to be specified in logics that allow one to make probabilistic assertions. The relation

between logic and probability has been studied a topic of interest since the inception of the field, namely due to the works of Boole, C. S. Peirce, Carnap, Burgess [Car50, Boo53, Bur69]. More recently, general-purpose probabilistic logics have been introduced [Nil86, FHM90, Ada96], which have a very large range of potential applications in formal verification of (uncertain) systems as well as in Artificial Intelligence. However, very few tools exist that in fact solve the satisfiability problem for such probabilistic logics. This is the aim of the second part of this work. In fact, we begin by developing the satisfiability procedure for a fragment of the probabilistic logic of Fagin et al. [FHM90], called GenPSAT, by a polynomial reduction to the theory of mixed integer linear programming. Having proved the correctness of the reduction we proceed to the implementation of a tool which solves this problem. With this implementation, we are able to study the phase transition behaviour of this specific problem, a phenomenon conjectured to be characteristic of NP-complete problems.

Figure 1 Inclusion diagram of several fragments of the probabilistic logic



In Figure 1, the relation between fragments of the probabilistic logic [FHM90] is laid out and a research path is clear. Having developed the satisfiability solver for GenPSAT, we are naturally at an arms reach of developing a satisfiability procedure for the full logic of Fagin et al. To do this, we find a polynomial reduction of this satisfiability problem to SMT, under the theory of quantifier-free linear integer and real arithmetic, a natural setting for this problem. We proceed the work by proving the correctness of the proposed reduction and building a tool which implements the reduction. We should remark that this tool, which extends the scope of application to the full language of the logic by Fagin et al. is much more efficient than the previously presented tool. Again, a phase transition study is made regarding the problem at hand and finally, some applications of the probabilistic satisfiability are presented in the area of information security, as well as the problem specifications in the language of the tool. In particular, we provide one example related to hardware verification under faulty gates and one example

which studies the effectiveness of Boolean masking against side-channel attacks.

We quickly realize how naturally side-channel problems are modelled in this probabilistic formalism and decide to take a step further in applications to the study and modelling of side-channel attacks. Side-channel attacks happen when an attacker is able to obtain supposedly private data through information that the system leaks via physical channels such as timing data [Koc96], power consumption [KJJ99], electromagnetic radiation [GMO01, QS01, AARR03], temperature [HS13] to name some impactful side channels.

Despite the security guarantees of the protocols, often enough, their security proofs do not usually take into account the information leaks the system may have through physical properties. In the traditional cryptography view of the world, an attacker only observes the public part of the protocol. For instance, in a private-key encryption scenario, Eve, an attacker, would only have access to the encryption of the message exchanged between Alice and Bob.

However, in the real-world, the encryption process takes time, the machine in which the encryption is being made consumes power, emits heat, electromagnetic radiation and sound. Unless defensive measures are taken into account more than one of these channels of physical information about the system may be available to an attacker. Furthermore, the attacker can actively force information leaks by injecting or forcing faults in the system. These fault attacks, introduced in [BDL97] and extended in [BDL01], can often lead to full secret recovery, even on standard ciphers such as AES [DLV03].

There are usually some approaches to thwart these attacks: on one hand there can be a physical shielding of the device running the cryptographic functions, trying to prevent unwanted leakage of information; on the other hand, there can be a logical shielding of the secrets, e.g., by means of a random mask which is applied to values during the execution of the algorithm [CJRR99, PR13]. We study the problem of deciding whether a system is perfectly masked, making use of the probabilistic formalism and respective satisfiability procedure presented in the previous chapters. In particular, we model the problem of deciding whether a formula is perfectly masked against a passive attacker in GGenPSAT. We proceed to provide a characterization of attackers with side-channel capabilities in this probabilistic formalism. With these notions, we are able formalize the notion of a perfectly masked circuit against such attackers. Surprisingly, when facing a very powerful attacker, this problem actually becomes easier and is shown to be in co-NP.

Overview

We now provide a more detailed description of each studied topic in this thesis. Furthermore, we also describe the main research goals that were taken into account

when studying each subject, as well as the main results obtained.

Many-Sorted Equivalence of Shiny and Strongly Polite Theories

With the development of satisfiability procedures for several different first-order theories, it became necessary to understand how to combine them in a modular way. Around 1979, Nelson and Oppen devised the first method to combine satisfiability procedures of disjoint and stably infinite theories into one satisfiability procedure of the union of those theories [NO79, Opp80]. After this, several generalizations and extensions of the method were made to encompass both non-stably infinite theories [TZ05, RRZ05, JB10a], as well as non-disjoint theories [DKR94, Zar02, TR03, CFR14a]. Two of those classes of theories are the shiny theories which were developed under the one-sorted case [TZ05], and strongly polite theories developed under the many-sorted setting [RRZ05, JB10a]. Each of these classes of theories allow for the combination of their satisfiability procedure with the satisfiability procedure of a disjoint arbitrary theory. However, the relation between these classes was not fully understood [JB10a, CR13], specially in the many-sorted case, and this is the main goal of the section.

Research goal: study Nelson-Oppen style combination procedures of first-order many-sorted theories namely

- study the relationship between shininess and strong politeness in the many-sorted case;
- devise a combination procedure for many-sorted shiny theories.

The achieved results are described below.

Main results:

- extension of the notion of shiny theories to the many-sorted case;
- proof of the equivalence, in the many-sorted case, of shiny and strongly polite theories;
- present two methods for the combination of many-sorted disjoint theories and shiny theories, one taking advantage of the equivalence between shiny and strongly polite theories and a direct method, which extends the combination method for one-sorted shiny theories.

The content of Chapter 1 is joint work with João Rasga. These results culminated in the following publication:

- [CR17] Casal, F. & Rasga, J. *Many-Sorted Equivalence of Shiny and Strongly Polite Theories* **J Autom Reasoning**. doi:10.1007/s10817-017-9411-y, 2017

Generalized Probabilistic Satisfiability

With the success that classical propositional logic has in modelling systems and their properties, it is only natural to think of extensions to it, namely to allow the ability to reason quantitatively about propositional formulas. The connections between logic and probabilities was in fact studied in the inception of logic as a field by Boole and others [Car50, Boo53, Bur69], and still plays a major role in several areas of knowledge, belief and automatic reasoning.

In 1990, Fagin et al. introduced a probabilistic logic [FHM90] that is able to reason with Boolean combinations of linear inequalities involving probabilities of propositional formulas. This logic has been very influential but until recently, there was no tool that solved its satisfiability procedure, even for smaller fragments of the logic. This changed with two solvers: PSAT [FB11, FB15] and GPSAT [BCF15] which are able to decide the satisfiability of two fragments of the probabilistic logic. These fragments, however, do not deal with linear inequalities of probabilistic terms, which is what we aim to solve.

Research goal: study a generalization of the PSAT fragment of Fagin et al. logic, GenPSAT, that allows linear combinations of probabilistic terms, namely

- study the computational complexity of this problem;
- devise a tool to solve the satisfiability problem of this fragment, and
- study the phase transition behaviour of this problem.

The main contributions are described below:

Main results:

- the satisfiability problem inherits the NP-completeness from the satisfiability problem for the logic of Fagin et al.
- development of a polynomial reduction from a GenPSAT instance to Mixed-integer linear programming;
- development of a tool that solves the satisfiability problem for GenPSAT, using the mentioned reduction;
- detection and study of phase transition behaviours for this problem.

The content of this chapter is joint work with Andreia Mordido and Carlos Caleiro. Furthermore, Chapter 2 is published in:

- [CCM17b] Caleiro, C. & Casal, F. & Mordido, A. *Generalized Probabilistic Satisfiability* **Electronic Notes in Theoretical Computer Science 332C**, pp. 39-56, 2017

Besides the publication, the satisfiability procedure was implemented, is open-source, and available at:

- [CMC16a] `https://github.com/fcasal/genpsat`

Classical Generalized Probabilistic Satisfiability

The work on this section follows naturally from the previously developed work on GenPSAT. Here, we aim to extend the GenPSAT fragment with Boolean combinations of linear inequalities involving probabilities of propositional formulas. This satisfiability problem has exactly the same expressiveness of the full logic by Fagin et al [FHM90].

Research goal: study a generalization of GenPSAT, GGenPSAT, which allows for Boolean combinations of linear inequalities involving probabilities of propositional formulas, namely

- study the computational complexity of this problem
- build a tool that solves the satisfiability problem of this fragment, and
- study the phase transition behaviour of this problem.

The main obtained results were the following:

Main results:

- as in GenPSAT, the satisfiability problem of GGenPSAT inherits the NP-completeness from the satisfiability problem for the logic of Fagin et al.
- development of a polynomial reduction from a GGenPSAT instance to an SMT problem, under the theory of quantifier-free linear integer and real arithmetic;
- development of a tool that solves the satisfiability problem for GGenPSAT using the above reduction;
- detection and study of the phase transition behaviour for this problem;
- provide two meaningful examples of applications of the formalism and the tool to problems in information security, namely hardware verification and side-channel attacks.

The content of this chapter is joint work with Andreia Mordido and Carlos Caleiro. Furthermore, Chapter 3 is published in:

- [CCM17a] Caleiro, C. & Casal, F. & Mordido, A. *Classical Generalized Probabilistic Satisfiability* **Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence**, IJCAI-17, pp. 908–914, 2017

Besides the publication, the satisfiability procedure was implemented, is open-source, and available at:

- [CMC16b] `https://github.com/fcasal/ggenpsat`

A Probabilistic Formalization of Attackers with Side-Channel Capabilities

In this section, we apply the probabilistic formalism and satisfiability solvers developed in the previous chapter to a problem in cryptography, specifically on the mitigation of side-channel attacks. In particular, we study the problem of determining whether a Boolean formula is perfectly masked. We start by modelling this problem in the probabilistic formalism and proceed to characterize several types of attackers with side-channel capabilities. Furthermore, we generalize the problem of deciding if a Boolean formulas is perfectly masked in the presence of these active attackers.

Research goal: study the problem of deciding if a system is perfectly masked against power-related side-channel attacks. In particular, we aim to

- study the computational complexity of the problem of deciding whether a Boolean formula is perfectly masked;
- model attackers with side-channel capabilities in the context of the probabilistic formalism;
- generalize the problem of perfect masking to encompass active attackers which have fault-injection capabilities;
- study the computational complexity of the generalized problems.

The main obtained results were the following:

Main results:

- model the problem of deciding whether a formula is perfectly masked in the probabilistic formalism as a GGenPSAT satisfiability problem;
- characterization of active attackers with side-channel capabilities and the definition of the generalized problem of perfect masking under these attackers;

- provide conditions under which the problem of deciding if a formula is perfectly masked against an active attacker belongs to co-NP.

The content of this chapter is joint work with Andreia Mordido and Carlos Caleiro and has been submitted for publication:

- [CCM17c] Caleiro, C. & Casal, F. & Mordido, A. *Generalized Probabilistic Satisfiability and Applications to Modelling Attackers with Side-Channel Capabilities*, submitted for publication, 2017

List of Publications

- [CR17]: Casal, F. & Rasga, J. *Many-Sorted Equivalence of Shiny and Strongly Polite Theories* **J Autom Reasoning**. doi:10.1007/s10817-017-9411-y, 2017
- [CCM17b]: Caleiro, C. & Casal, F. & Mordido, A. *Generalized Probabilistic Satisfiability* **Electronic Notes in Theoretical Computer Science 332C**, pp. 39-56, 2017
- [CCM17a]: Caleiro, C. & Casal, F. & Mordido, A. *Classical Generalized Probabilistic Satisfiability* **Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence**, IJCAI-17, pp. 908–914, 2017
- [CCM17c] Caleiro, C. & Casal, F. & Mordido, A. *Generalized Probabilistic Satisfiability and Applications to Modelling Attackers with Side-Channel Capabilities*, submitted for publication, 2017

List of Tools

In the context of this thesis, two satisfiability solvers for probabilistic logics were developed and are open source:

- [CMC16a] `https://github.com/fcasal/genpsat`
- [CMC16b] `https://github.com/fcasal/ggenpsat`

Chapter 1

Many-Sorted Equivalence of Shiny and Strongly Polite Theories

1.1 Introduction

The Nelson-Oppen method is a well-known method for modularly combining satisfiability procedures of given theories. The method was proposed by Nelson and Oppen in 1979, [NO79], and provides a way of deciding the satisfiability of quantifier-free formulas in the union of two (one-sorted) theories, as long as both of them have their own procedure for deciding the satisfiability problem of quantifier-free formulas. After a correction, see [Opp80], the two main conditions of the Nelson-Oppen[1] method are that:

- the theories are *stably infinite*,
- their signatures are disjoint.

Concerned about the fact that many theories of interest, such as those admitting only finite models, are not stably infinite, Tinelli and Zarba, in [TZ05], showed that the Nelson-Oppen combination procedure still applies when the stable infiniteness condition is replaced by the requirement that all but one of the theories are *shiny*. However, a shiny theory must be equipped with a function that computes minimal cardinalities of models of formulas, which is inherently hard to compute.

In order to overcome the problem of computing this function and in proving that a theory is shiny, as well as to generalize these combination methods to the

[1] A correctness proof of the method was presented by Tinelli and Harandi in [TH96].

many-sorted case, Ranise, Ringeissen and Zarba proposed an alternative requirement, *politeness*, in [RRZ05], and analysed its relationship with shininess. A polite theory has to be equipped with a witness function, which was thought to be easier to compute than the mincard function of shiny theories. They showed that given a polite theory and an arbitrary one, the Nelson-Oppen combination procedure is still valid when the theories have disjoint signatures and both have their own procedure for deciding the satisfiability problem of quantifier-free formulas. Some time later, in [JB10a], Jovanović and Barrett reported that the politeness notion provided in [RRZ05] allowed, after all, witness functions that are not sufficiently strong to prove the combination theorem. To overcome this issue they provided a stronger notion of politeness, in the sequel called *strong politeness*, equipped with a stronger witness function, s-witness, that allowed to prove the combination theorem. However, in [JB10a], the relationship between strong politeness and shininess was not studied. This motivated the work in [CR13], where the authors investigated the relationship between shiny and strongly polite theories in the one-sorted case. They showed that a shiny theory with a decidable quantifier-free satisfiability problem is strongly polite, and, for the other direction, they provided two different sets of conditions under which a polite theory is shiny.

In [Fon09], Fontaine introduced *gentle* theories that are a natural generalization of shiny theories. In [Fon09] and [AF11], the authors further showed that the union of gentle theories is gentle, and classified several theories in decidable fragments of first-order logic in terms of gentleness and shininess. Gentle theories can be combined with a very broad class of theories, although there is no Nelson-Oppen theorem for the combination of gentle theories with an arbitrary theory. Furthermore, gentle theories have also played a role on the area of non-disjoint combination of theories [CFR14b].

Herein we settle the loose end from [CR13] and show that the class of many-sorted shiny theories coincides, with respect to any set of sorts, with the class of strongly polite theories, when the theory is equipped with a quantifier-free satisfiability solver. We begin by adapting the notion of shininess to the many-sorted case. This adaptation is by no means immediate. For example, the stably finite notion, when adapted to the many-sorted case, has to include a condition on the cardinality of the finite domains and the mincard function had to be replaced by a more general function, minmods, that returns tuples of the cardinalities of the *minimal* models of the theory that satisfy a given formula. Nonetheless these generalized notions coincide with the usual ones that they extend when seen in the one-sorted case. Then, we extend the results in [CR13] in a two-fold manner: on one hand we prove the equivalence between shiny and strongly polite theories unconditionally, and on the other hand we obtain this result in the many-sorted context. These results do not have the restriction imposed in [RRZ05] when relating polite theories with shiny theories, that the set of sorts has to be the full set of sorts in the signature. Capitalizing on this equivalence and on the Nelson-

Oppen combination theorem for many-sorted strongly polite theories in [JB10a], we establish a Nelson-Oppen combination theorem for many-sorted shiny theories.

1.1.1 Organization of the Chapter

The chapter is organized as follows: in Section 1.2 we introduce the main notions and definitions used throughout the chapter. In Section 1.3 we show that when equipped with a quantifier-free satisfiability solver, the classes of many-sorted shiny and strongly polite theories coincide for any given finite set of sorts. In Section 1.4 we capitalize on the proved equivalence between strongly polite and shiny, and on the combination theorem for strongly polite theories and establish the Nelson-Oppen combination theorem for many-sorted shiny theories. In Section 1.5 we conclude the chapter and provide some directions for further research.

1.2 Preliminaries

The results in this chapter concern theories of many-sorted first-order logic with equality. For each sort, we assume a disjoint countably infinite set of variables. We follow the many-sorted presentation of first-order logic with equality as is done in [End01].

1.2.1 Syntax

A *signature* is a tuple $\Sigma = \langle \Sigma^S, \Sigma^F, \Sigma^P, \alpha, \tau \rangle$ where Σ^S is the non-empty finite set of sorts, Σ^F is the set of function symbols, Σ^P is the set of predicate symbols, α is a map that for each function and predicate symbol returns its arity and τ is a map that for each function and predicate symbol returns its type. When applied to variables or sets of variables, τ returns the sorts of the variables. For each sort $\sigma \in \Sigma^S$, we use $\cong_\sigma$ to denote the equality logic symbol over pairs of terms of sort σ and assume the standard many-sorted definitions of Σ-*atom* and Σ-*term*. A Σ-*formula* is inductively defined as usual over Σ-atoms using the connectives $\wedge, \vee, \neg, \rightarrow$ or the quantifiers $\forall$ and $\exists$. We denote by $\mathsf{QF}(\Sigma)$ the set of Σ-formulas with no occurrences of quantifiers and, given a Σ-formula φ, by $\mathsf{vars}(\varphi)$ the set of free variables of φ, i.e., the set of variables not under the scope of a quantifier. Furthermore, we denote by $\mathsf{vars}_\sigma(\varphi)$ the set of free variables of sort σ occurring in φ. Given a set of terms T and a sort σ, we denote by T_σ the set of terms in T of sort σ, and say that a Σ-formula is a Σ-*sentence* if it has no free variables. In the sequel, when there is no ambiguity, we may omit the reference to the signature when referring to atoms, terms, formulas and sentences.

Given a finite set of variables Y over a set of sorts S and $E \subseteq Y^2$, we write

$$E \sqsubseteq Y^2$$

to denote that E is a family of sort-wise equivalence relations over Y, i.e.,

$$E = \bigcup_{\sigma \in S} E_\sigma,$$

and for each sort $\sigma \in S$, E_σ is an equivalence relation on Y_σ^2.

Definition 1.1 (Arrangement formula)**.** *Given a finite set of variables Y over a set of sorts S and $E \sqsubseteq Y^2$, the* arrangement formula *induced by E over Y, denoted by*

$$\delta_E^Y$$

is the formula

$$\bigwedge_{\sigma \in S} \delta_{E_\sigma}^{Y_\sigma}$$

where $\delta_{E_\sigma}^{Y_\sigma}$ is

$$\bigwedge_{(x,y) \in E_\sigma} (x \cong_\sigma y) \wedge \bigwedge_{(x,y) \in Y_\sigma^2 \setminus E_\sigma} \neg (x \cong_\sigma y) \ .$$

In the sequel, when there is no ambiguity, we may simply denote δ_E^Y by δ_E.

1.2.2 Semantics

Given a signature Σ, a Σ-*interpretation* $\mathcal{A}$ over a set of variables X is a map that interprets:

- each sort $\sigma \in \Sigma^S$ as a non-empty set A_σ;
- each variable $x \in X$ with sort σ as an element $x^\mathcal{A} \in A_\sigma$;
- each function symbol $f \in \Sigma^F$ of arity n and type $\sigma_1 \times \ldots \times \sigma_n \to \sigma$ as a map $f^\mathcal{A} : A_{\sigma_1} \times \ldots \times A_{\sigma_n} \to A_\sigma$, and
- each predicate symbol $p \in \Sigma^P$ of arity n and type $\sigma_1 \times \ldots \times \sigma_n$ as a subset $p^\mathcal{A}$ of $A_{\sigma_1} \times \ldots \times A_{\sigma_n}$.

We denote the domain of a Σ-interpretation $\mathcal{A}$ by A, i.e., the collection of the domains A_σ for each sort σ. In the sequel, when there is no ambiguity, we may omit the reference to the signature when referring to interpretations.

Given an interpretation $\mathcal{A}$ and a term t, we denote by $t^\mathcal{A}$ the interpretation of t under $\mathcal{A}$. Similarly, we denote by $\varphi^\mathcal{A}$ the truth value of the formula φ under

the interpretation $\mathcal{A}$. Furthermore, given a set T of terms, we denote by $[\![T]\!]^{\mathcal{A}}$ the set $\{t^{\mathcal{A}} : t \in T\}$. Finally, we write $\mathcal{A} \models \varphi$ when the formula φ is true under the interpretation $\mathcal{A}$, i.e., $\mathcal{A}$ satisfies φ.

A formula φ is *satisfiable* if it is true under some interpretation. It is *unsatisfiable* otherwise.

Given sets of variables Y and X, we say that two interpretations $\mathcal{A}$ and $\mathcal{B}$ over X are *equivalent modulo* Y whenever $A = B$, $f^{\mathcal{A}} = f^{\mathcal{B}}$ for each function symbol f, $p^{\mathcal{A}} = p^{\mathcal{B}}$ for each predicate symbol p, and $x^{\mathcal{A}} = x^{\mathcal{B}}$ for each variable x in $X \setminus Y$.

We also say that an *interpretation $\mathcal{A}$ is finite (resp. infinite)* with respect to a set S of sorts when, for each sort $\sigma \in S$, the set A_σ is finite (resp. infinite).

1.2.3 Theories

Given a signature Σ, a Σ-*theory* is a set of Σ-sentences, and given a Σ-theory $\mathcal{T}$, a $\mathcal{T}$-*model* is a Σ-interpretation that satisfies all the sentences of $\mathcal{T}$. A formula φ is $\mathcal{T}$-*satisfiable* when there is a $\mathcal{T}$-model that satisfies it, and two formulas are $\mathcal{T}$-*equivalent* if they are interpreted to the same truth value in every $\mathcal{T}$-model. Given a Σ_1-theory $\mathcal{T}_1$ and a Σ_2-theory $\mathcal{T}_2$, their union, $\mathcal{T}_1 \cup \mathcal{T}_2$, is a $\Sigma_1 \cup \Sigma_2$-theory defined by the union of the sentences of $\mathcal{T}_1$ with the sentences of $\mathcal{T}_2$. In the sequel, when there is no ambiguity, we may omit the reference to the signature when referring to theories.

Definition 1.2 (Smoothness, [RRZ05])**.** *A Σ-theory $\mathcal{T}$ is* smooth *with respect to a set of sorts $S \subseteq \Sigma^S$ if for every quantifier-free formula φ, $\mathcal{T}$-model $\mathcal{A}$ satisfying φ and cardinals $\kappa_\sigma \geq |A_\sigma|$ for each $\sigma \in S$, there exists a $\mathcal{T}$-model $\mathcal{B}$ satisfying φ such that $|B_\sigma| = \kappa_\sigma$ for all $\sigma \in S$.*

Definition 1.3 (Strong finite witnessability, [JB10a])**.** *A Σ-theory $\mathcal{T}$ is* strongly finitely witnessable *with respect to a set of sorts $S \subseteq \Sigma^S$ if there exists a computable function* $\mathsf{s\text{-}witness} : \mathsf{QF}(\Sigma) \to \mathsf{QF}(\Sigma)$ *such that for every quantifier-free formula φ the following conditions hold:*

- *φ and $\exists \vec{w}\ \mathsf{s\text{-}witness}(\varphi)$ are $\mathcal{T}$-equivalent, where $\vec{w}$ are the variables in the formula $\mathsf{s\text{-}witness}(\varphi)$ which do not occur in φ;*
- *for every finite set Y of variables[2] with sorts in S and $E \sqsubseteq Y^2$, if $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ is $\mathcal{T}$-satisfiable then there exists a $\mathcal{T}$-model $\mathcal{A}$ of $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ such that $A_\sigma = [\![\mathsf{vars}_\sigma\left(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y\right)]\!]^{\mathcal{A}}$, for all $\sigma \in S$.*

A function satisfying the above properties is called a *strong witness function for $\mathcal{T}$* with respect to S.

[2] As in [JB10a], we do not restrict Y to be the set of variables in s-witness since this generality is needed to show Lemma A.2 and Theorem 3.7 of [JB10a].

Definition 1.4 (Strong politeness, [JB10a]). *A Σ-theory is* strongly polite *with respect to a set of sorts $S \subseteq \Sigma^S$ whenever it is smooth and strongly finitely witnessable with respect to S.*

Shiny theories were introduced by Tinelli and Zarba in [TZ05] and extended to the many-sorted case by Ranise, Ringeissen and Zarba in [RRZ05]. In the one-sorted case, these theories are characterized by having a computable function which given a satisfiable formula returns the cardinality of the smallest model of the theory that satisfies the formula. However, when we are dealing with many-sorted theories, unless orderings are imposed on sorts, there is more than one minimal model. By minimal models, we refer to the models that cannot be spanned by smoothness from other models. In a sense, they form a basis to the set of models of smooth theories.

Here, instead of defining the cardinality of the smallest model in terms of some measure as in [RRZ05], where models are compared by their maximum cardinality, we consider the cardinalities of minimal models.

Given a tuple $\bar{k}$ indexed by a set of sorts S, we denote by $\bar{k}[\sigma]$ its σ-component for each $\sigma \in S$.

Definition 1.5 ($\mathsf{minmods}$ function). *Given a Σ-theory $\mathcal{T}$, a set of sorts $S \subseteq \Sigma^S$ and a $\mathcal{T}$-satisfiable quantifier free formula φ, $\mathsf{minmods}_{\mathcal{T},S}(\varphi)$ is the set of S-tuples defined as follows:*

$\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ *iff*

- *there is a model $\mathcal{A}$ of $\mathcal{T} \cup \{\varphi\}$ with $|A_\sigma| = \bar{k}[\sigma]$ for each $\sigma \in S$;*
- *for each model $\mathcal{B}$ of $\mathcal{T} \cup \{\varphi\}$ with $\langle |B_\sigma| \rangle_{\sigma \in S} \neq \bar{k}$, there is a sort $\sigma \in S$ such that $\bar{k}[\sigma] < |B_\sigma|$.*

We also provide a notion of many-sorted stable finiteness different from the one in [RRZ05].

Definition 1.6 (Stable finiteness). *A Σ-theory $\mathcal{T}$ is* stably finite *with respect to a set of sorts $S \subseteq \Sigma^S$ if for every quantifier-free formula φ satisfied by a $\mathcal{T}$-model $\mathcal{A}$ there exists a finite $\mathcal{T}$-model $\mathcal{A}'$ with respect to S satisfying φ such that $|A'_\sigma| \leq |A_\sigma|$ for each $\sigma \in S$.*

Notice how this formulation of many-sorted stable finiteness coincides with the usual stable finiteness property on one-sorted theories. Furthermore, with this formulation we guarantee that, for a stably finite theory, the set $\mathsf{minmods}_{\mathcal{T},S}(\varphi)$ only contains tuples with finite cardinalities. Observe that this is not the case with the many-sorted extension of the stable finiteness notion in [RRZ05], which states that if a formula has a model then it has a finite model.

As an example of a theory stably finite according to [RRZ05] (but not stably finite according to the notion proposed in Definition 1.6), with minimal models of infinite cardinalities, consider a two-sorted theory that accepts all models $\mathcal{A}$ with cardinalities such that

$$\text{either } |A_1| \geq 2 \text{ and } |A_2| = \infty \text{ or } |A_1| \geq 3 \text{ and } |A_2| \geq 3 \ .$$

Then, in this theory and with the notion of stable finiteness of [RRZ05], the formula $x \cong_1 x$, has a minimal model with an infinite component, $\langle 2, \infty \rangle$.

We now show that the minmods set for a given formula is finite, by checking that there can only be a finite number of finite tuples that satisfy the second property of the minmods notion.

Proposition 1.1. *Let $\mathcal{T}$ be a many-sorted stably finite Σ-theory with respect to a set of sorts $S \subseteq \Sigma^S$. For any $\mathcal{T}$-satisfiable quantifier-free formula φ, the set $\mathsf{minmods}_{\mathcal{T},\mathcal{S}}(\varphi)$ is finite and only contains tuples with finite cardinalities.*

Proof. Observe that by the second property of the minmods function, for any two different elements $\bar{k}_1, \bar{k}_2 \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$, there exists a sort σ such that $\bar{k}_1[\sigma] < \bar{k}_2[\sigma]$. This also holds for the other direction, i.e., there exists a sort σ' such that $\bar{k}_2[\sigma'] < \bar{k}_1[\sigma']$. Thus, the elements in minmods are incomparable in the product order, and as such, we can bound the size of the minmods set by the size of the largest set of incomparable elements in the product order in $\mathbb{N}^{|S|}$ (we only need to take into account finite cardinalities because the theory is stably finite). It happens that there are no infinite sets of incomparable elements in the product order in $\mathbb{N}^{|S|}$, and so the minmods set is finite.

The fact that there are no infinite sets of incomparable elements in the product order in $\mathbb{N}^{|S|}$ can be restated in terms of partial order theory[3]: there are no infinite antichains (sets of incomparable elements) in the partial ordered set $\mathbb{N}^{|S|}$. Take $|S| = k$ and view an element $\langle a_1, \ldots, a_k \rangle \in \mathbb{N}^k$ as the monomial $x_1^{a_1} \cdots x_k^{a_k}$ in the polynomial ring $\mathbb{Z}_2[x_1, \ldots, x_k]$. Suppose there is an infinite antichain, and consider the ideal generated by its elements. By Hilbert's Basis theorem [Hil70, CLO07], this ideal is finitely generated. Since all elements generated by each basis monomial are comparable to it, the only possible incomparable elements are the basis elements, which are in finite number. Thus there are no infinite antichains. □

In the sequel, we use the following useful lemmas:

Lemma 1.1. *For each $\mathcal{T}$-satisfiable quantifier-free formula φ, $\mathsf{minmods}_{\mathcal{T},S}(\varphi) \neq \emptyset$.*

[3] We recently learned that this result is known as Dickson's lemma [Dic13].

Proof. Let $\mathcal{A}$ be a $\mathcal{T}$-model of φ. Then, either $\langle |A_\sigma| \rangle_{\sigma \in S}$ is in $\mathsf{minmods}_{\mathcal{T},S}(\varphi)$ or it is not and so there is another $\mathcal{T}$-model of φ smaller than $\mathcal{A}$ which is in $\mathsf{minmods}_{\mathcal{T},S}(\varphi)$. In either case, the set is not empty. □

Lemma 1.2. *Given a many-sorted stably finite Σ-theory $\mathcal{T}$ with respect to a set of sorts $S \subseteq \Sigma^S$ and a $\mathcal{T}$-satisfiable quantifier free formula φ, for any $\mathcal{T}$-model $\mathcal{A}$ of φ, there is a tuple $\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ such that*

$$\bar{k}[\sigma] \leq |A_\sigma|, \text{ for all } \sigma \in S \ .$$

Proof. Consider the following set M of tuples of S-cardinalities of finite $\mathcal{T}$-models of φ:

$$\{\langle |B_\sigma| \rangle_{\sigma \in S} : \mathcal{B} \text{ is a finite } \mathcal{T}\text{-model of } \varphi \text{ and } |B_\sigma| \leq |A_\sigma| \text{ for all } \sigma \in S\} \subseteq \mathbb{N}^{|S|} \ .$$

This set is non-empty due to the stable finiteness of $\mathcal{T}$. Consider the product order over $\mathbb{N}^{|S|}$ defined as usual, i.e., $\bar{k} \leq \bar{k}'$ iff $\bar{k}[\sigma] \leq \bar{k}'[\sigma]$ for all $\sigma \in S$. Since $\mathbb{N}^{|S|}$ is lower-bounded, and M is a non-empty subset of it, we conclude that M must have a minimal element $\bar{m}$, that is, an element which has no smaller element than it. We claim that $\bar{m} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$: if it were not, by definition of $\mathsf{minmods}$ and since $\bar{m}$ corresponds to a tuple of cardinalities of a $\mathcal{T}$-model of φ, it would mean that there would exist a model $\mathcal{C}$ such that for all $\sigma \in S$, $|C_\sigma| \leq \bar{m}[\sigma]$ and $\langle |C_\sigma| \rangle_{\sigma \in S} \neq \bar{m}$. But this contradicts the fact that $\bar{m}$ is a minimal element of M. □

Equipped with the notions of smoothness, stable finiteness, as well as the $\mathsf{minmods}$ function (which plays the role of the $\mathsf{mincard}$ function in the one-sorted case) we are able to define shininess in the many-sorted case. We emphasize that each of these notions coincide with their original notions in the one-sorted case as introduced in [TZ05].

Definition 1.7 (Shininess)**.** *A Σ-theory $\mathcal{T}$ is* shiny *with respect to a set of sorts $S \subseteq \Sigma^S$ whenever it is smooth and stably finite with respect to S, and the function $\mathsf{minmods}_{\mathcal{T},S}$ is computable.*

1.3 Shiny and Strongly Polite Theories

In this section we analyse the relationship between many-sorted shiny and strongly polite theories. We start by showing that a shiny theory with respect to a set of sorts is strongly polite with respect to the same set, assuming that the theory has a decidable quantifier-free satisfiability problem.

Proposition 1.2. *A shiny theory with respect to a set of sorts S with a decidable quantifier-free satisfiability problem is strongly polite with respect to S.*

Proof. Let $\mathcal{T}$ be a shiny Σ-theory with respect to a set $S \subseteq \Sigma^S$ of sorts and Sat an algorithm that solves its quantifier-free satisfiability problem. Since a shiny theory is by definition smooth, in order to conclude that $\mathcal{T}$ is strongly polite with respect to S, we are left to prove that $\mathcal{T}$ is strongly finitely witnessable with respect to S. In the sequel, given a $\mathcal{T}$-satisfiable quantifier-free formula φ and a family of equivalence relations $E \sqsubseteq \mathsf{vars}(\varphi)^2$, if $\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}$ is $\mathcal{T}$-satisfiable, we denote by $\mathsf{MM}(\varphi, E)$ the set $\mathsf{minmods}_{\mathcal{T},S}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)$.

In other words, $\mathsf{MM}(\varphi, E)$ has the tuples of cardinalities of the minimal $\mathcal{T}$-models that satisfy $\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}$. Given a tuple $\bar{k} \in \mathsf{MM}(\varphi, E)$, we denote the cardinality of the σ domain by $\bar{k}[\sigma]$. Observe that all the $\bar{k}[\sigma]$ are finite due to the stable finiteness of $\mathcal{T}$ with respect to S.

The proof is structured in the following manner: we begin by proposing a strong witness function $\mathsf{s\text{-}witness}$ and in Lemma 1.3 we show that it is computable. In Lemma 1.4 we show that this function satisfies the first condition of strong finite witnessability and finally in Lemma 1.5 that it satisfies the second condition.

Let $\mathsf{s\text{-}witness} : \mathsf{QF}(\Sigma) \to \mathsf{QF}(\Sigma)$ be such that

$$\mathsf{s\text{-}witness}(\varphi) = \varphi \wedge \Omega,$$

where Ω is

$$\bigwedge_{\substack{E \sqsubseteq \mathsf{vars}(\varphi)^2 \\ \mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)=1}} \left(\delta_E^{\mathsf{vars}(\varphi)} \to \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]} \right)$$

and $\gamma_{\bar{k}[\sigma]}$ is

$$\begin{cases} \bigwedge_{\substack{i,j=1 \\ i \neq j}}^{\bar{k}[\sigma]} \neg(w_{i,\sigma} \cong_\sigma w_{j,\sigma}), & \text{if } \bar{k}[\sigma] > 1 \\ (w_{1,\sigma} \cong_\sigma w_{1,\sigma}), & \text{if } \bar{k}[\sigma] = 1 \end{cases}$$

and $w_{1,\sigma}, \ldots, w_{\bar{k}[\sigma],\sigma}$ are distinct σ-variables not occurring in φ and in $\gamma_{\bar{k}'[\sigma]}$ for all $\bar{k}' \neq \bar{k}$.

Lemma 1.3. *The function* $\mathsf{s\text{-}witness}$ *is a computable function.*

Proof. It is immediate to conclude that $\mathsf{s\text{-}witness}$ is computable since:

- there is a finite number of sets E with $E \sqsubseteq \mathsf{vars}(\varphi)^2$ since $\mathsf{vars}(\varphi)$ is finite;
- the formula $\delta_E^{\mathsf{vars}(\varphi)}$ can be computed in a finite number of steps since E and $\mathsf{vars}(\varphi)$ are finite;
- the set $\mathsf{MM}\,(\varphi, E)$ is computable since the $\mathsf{minmods}$ function is computable and finite by Proposition 1.1;

- the formula $\gamma_{\bar{k}[\sigma]}$ is computable since $\bar{k}[\sigma]$ is a natural number due to the stable finiteness of $\mathcal{T}$ with respect to S.

□

We now show that s-witness satisfies the first condition of strong finite witnessability.

Lemma 1.4. *Let φ be a quantifier-free formula. Then, φ and $\exists\vec{w}$ $\mathsf{s\text{-}witness}(\varphi)$ are $\mathcal{T}$-equivalent.*

Proof. Let $\mathcal{A}$ be a $\mathcal{T}$-model. Assume that $\mathcal{A} \models \exists\vec{w}\ \mathsf{s\text{-}witness}(\varphi)$. Then $\mathcal{A} \models \varphi \wedge \exists\vec{w}\ \Omega$, and so $\mathcal{A} \models \varphi$.

For the other direction, assume $\mathcal{A} \models \varphi$. We need to show that

$$\mathcal{A} \models \exists\vec{w} \bigwedge_{\substack{E \sqsubseteq \mathsf{vars}(\varphi)^2 \\ \mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)=1}} \left(\delta_E^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]} \right) .$$

Let $\mathcal{A}'$ be an interpretation equivalent modulo $\vec{w}$ to $\mathcal{A}$ (and so with the same domain and the same interpretation of functions, predicates and of all variables except possibly those in $\vec{w}$) such that for each sort $\sigma \in S$:

- if A_σ is infinite then $w_{1,\sigma}^{\mathcal{A}'} \neq w_{2,\sigma}^{\mathcal{A}'}$ for every $w_{1,\sigma}, w_{2,\sigma} \in \vec{w}_\sigma$;
- if A_σ is finite then for each $E \sqsubseteq \mathsf{vars}(\varphi)^2$ with $\mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right) = 1$:
  - if $\bar{k}[\sigma] \leq |A_\sigma|$ then $w_i^{\mathcal{A}'} \neq w_j^{\mathcal{A}'}$ for every $w_i, w_j \in \mathsf{vars}(\gamma_{\bar{k}[\sigma]})$;
  - otherwise, $w_i^{\mathcal{A}'} = w_j^{\mathcal{A}'}$ for every $w_i, w_j \in \mathsf{vars}(\gamma_{\bar{k}[\sigma]})$.

Then

$$\mathcal{A}' \models \bigwedge_{\substack{E \sqsubseteq \mathsf{vars}(\varphi)^2 \\ \mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)=1}} \left(\delta_E^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]} \right)$$

since for each $E \sqsubseteq \mathsf{vars}(\varphi)^2$ with $\mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right) = 1$ either

- $\mathcal{A}' \not\models \delta_E^{\mathsf{vars}(\varphi)}$ and so $\mathcal{A}' \models \delta_E^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$; or

- $\mathcal{A}' \models \delta_E^{\mathsf{vars}(\varphi)}$. Observe that $\mathcal{A}' \models \varphi$ because $\mathcal{A}$ and $\mathcal{A}'$ may only differ in the interpretation of the variables in $\vec{w}$ which do not occur in φ. So $\mathcal{A}' \models \varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}$. Since $\mathcal{A}'$ is a model for $\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}$, by Lemma 1.2, there is a $\bar{k} \in \mathsf{MM}(\varphi, E)$ such that the cardinality of A_σ has to be greater or equal than $\bar{k}[\sigma]$ for each sort σ in S. Hence $\mathcal{A}' \models \gamma_{\bar{k}[\sigma]}$ for each $\sigma \in S$ and so $\mathcal{A}' \models \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$. Thus, $\mathcal{A}' \models \delta_E^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$.

□

We now show that s-witness satisfies the second condition of strong finite witnessability.

Lemma 1.5. *Let Y be a finite set of variables with sorts in S, φ a quantifier-free formula and $\psi = \mathsf{s\text{-}witness}(\varphi)$. Given a family $E \sqsubseteq Y^2$, if $\psi \wedge \delta_E^Y$ is $\mathcal{T}$-satisfiable, then there exists a $\mathcal{T}$-model $\mathcal{A}$ that satisfies $\psi \wedge \delta_E^Y$ such that $A_\sigma = [\![\mathsf{vars}_\sigma(\psi \wedge \delta_E^Y)]\!]^{\mathcal{A}}$, for each sort σ in S.*

Proof. Let Y be a finite set of variables with sorts in S and $E \sqsubseteq Y^2$ such that $\psi \wedge \delta_E^Y$ is $\mathcal{T}$-satisfiable. For each $\sigma \in S$, let p_σ be the number of equivalence classes induced by δ_E^Y, and $Y_1^\sigma, \ldots, Y_{p_\sigma}^\sigma$ those classes. These form a partition of the set of variables Y_σ. Furthermore, let $\mathcal{A}$ be a $\mathcal{T}$-model of $\psi \wedge \delta_E^Y$ and let $\delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$ be the arrangement formula induced by

$$E_\mathcal{A} = \bigcup_{\sigma \in S} \left\{ (x, y) : x, y \in \mathsf{vars}_\sigma(\varphi) \text{ and } x^\mathcal{A} = y^\mathcal{A} \right\} \quad .$$

Then, obviously, $\delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$ is satisfied by $\mathcal{A}$. Hence, $\varphi \wedge \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$ is $\mathcal{T}$-satisfiable and thus has a $\mathcal{T}$-model with cardinality tuple $\bar{k}$ in $\mathsf{MM}(\varphi, E_\mathcal{A})$ since $\mathcal{T}$ is stably finite with respect to S. Let $K_\sigma = \max\left\{\bar{k}[\sigma], p_\sigma\right\}$. By the smoothness of $\mathcal{T}$ and since $\varphi \wedge \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$ is $\mathcal{T}$-satisfiable, let $\mathcal{B}$ be a $\mathcal{T}$-model such that

$$\mathcal{B} \models \varphi \wedge \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)} \quad \text{and} \quad |B_\sigma| = K_\sigma \text{ for each sort } \sigma \in S,$$

and, for each sort σ in S, let $d_1^\sigma, \ldots, d_{p_\sigma}^\sigma$ be distinct elements of B_σ such that for $i = 1, \ldots, p_\sigma$

$$d_i^\sigma = y^\mathcal{B} \text{ for all } y \in Y_i^\sigma \cap \mathsf{vars}(\varphi),$$

and assuming that the variables of $\gamma_{\bar{k}[\sigma]}$ are $w_1, \ldots, w_{\bar{k}[\sigma]}$ let $e_1^\sigma, \ldots, e_{\bar{k}[\sigma]}^\sigma$ be distinct elements of B_σ such that

$$e_j^\sigma = d_i^\sigma \text{ if } w_j \in Y_i^\sigma$$

for $j = 1, \ldots, \bar{k}[\sigma]$ and $i = 1, \ldots, p_\sigma$. Observe that distinct variables in $w_1, \ldots, w_{\bar{k}[\sigma]}$ are in distinct sets in $Y_1^\sigma, \ldots, Y_{p_\sigma}^\sigma$ if they are in any Y_i^σ, since $\mathcal{A} \models \delta_E^Y$

and $\mathcal{A} \models \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$. Let $\mathcal{B}'$ be the $\mathcal{T}$-model equivalent modulo $(\vec{w} \cup (Y \setminus \mathsf{vars}(\varphi)))$ to $\mathcal{B}$ such that

$$x^{\mathcal{B}'} = \begin{cases} d_i^\sigma & \text{if } x \in Y_i^\sigma \text{ for some } i \in \{1, \ldots, p_\sigma\} \text{ and } \sigma \in S \\ e_j^\sigma & \text{if } x \notin Y \text{ and } x \text{ is } w_j \text{ with } w_j \in \mathsf{vars}\left(\gamma_{\bar{k}[\sigma]}\right) \text{ and } \sigma \in S \\ x^{\mathcal{B}} & \text{if } x \notin Y \text{ and } x \notin \mathsf{vars}\left(\gamma_{\bar{k}[\sigma]}\right) \text{ for all } \sigma \in S \end{cases}$$

for each $x \in \vec{w} \cup (Y \setminus \mathsf{vars}(\varphi))$. Let us now prove that $\mathcal{B}' \models \varphi \wedge \Omega \wedge \delta_E^Y$:
(a) $\mathcal{B}' \models \varphi$. This follows immediately taking into account that $\mathcal{B} \models \varphi$ and that $\mathcal{B}$ and $\mathcal{B}'$ may only differ in variables in $\vec{w} \cup (Y \setminus \mathsf{vars}(\varphi))$, hence not occurring in φ;
(b) $\mathcal{B}' \models \Omega$. Observe that $\mathcal{B}' \models \varphi \wedge \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$ since $\mathcal{B}$ and $\mathcal{B}'$ may only differ in variables in $\vec{w} \cup (Y \setminus \mathsf{vars}(\varphi))$, hence not occurring in $\varphi \wedge \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$. Moreover $\mathcal{B}' \models \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$ by definition, and so $\mathcal{B}' \models \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi, E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$. Since $\mathcal{B}' \models \delta_{E_\mathcal{A}}^{\mathsf{vars}(\varphi)}$, we have that $\mathcal{B}' \not\models \delta_E^{\mathsf{vars}(\varphi)}$ for all $E \neq E_\mathcal{A}$ with $E \sqsubseteq \mathsf{vars}(\varphi)^2$. Hence $\mathcal{B}' \models \delta_E^{\mathsf{vars}(\varphi)} \rightarrow \bigvee_{\bar{k} \in \mathsf{MM}(\varphi, E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]}$ for all $E \sqsubseteq \mathsf{vars}(\varphi)^2$ with $\mathsf{Sat}(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}) = 1$ and so $\mathcal{B}' \models \Omega$;
(c) $\mathcal{B}' \models \delta_E^Y$. We only need to verify that $\mathcal{B}'$ satisfies the equalities and disequalities induced by E. This holds since by construction, for each $\sigma \in S$, $\mathcal{B}'$ assigns the same value to variables in Y_i^σ, and assigns different values to variables in different sets Y_i^σ.

Finally it remains to show that $B'_\sigma = [\![\mathsf{vars}_\sigma(\varphi \wedge \Omega \wedge \delta_E^Y)]\!]^{\mathcal{B}'}$ for each sort $\sigma \in S$:
($\subseteq$): Let $d \in B'_\sigma$. Then d is either a d_i^σ for some $i = 1, \ldots, p_\sigma$ or a e_j^σ for some $j = 1, \ldots, \bar{k}[\sigma]$. In the case that $d = d_i^\sigma$ then we have that $d = x^{\mathcal{B}'}$ for all $x \in Y_i^\sigma$. On the other hand, if $d = e_j^\sigma$ then $d = w_j^{\mathcal{B}'}$ for the w_j variable in $\mathsf{vars}_\sigma(\gamma_{\bar{k}[\sigma]})$;
($\supseteq$): Obviously, $[\![\mathsf{vars}_\sigma(\varphi \wedge \Omega \wedge \delta_E^Y)]\!]^{\mathcal{B}'} \subseteq B'_\sigma$ by definition. □

Combining Lemmas 1.3, 1.4 and 1.5 we conclude that a shiny theory with respect to a set S is strongly finitely witnessable with respect to S, hence strongly polite with respect to S since it is smooth by definition. □

We find relevant to remark that the given proof is constructive in the way that we provide a way to build a strong witness function for a shiny theory, provided that it has a decidable quantifier-free satisfiability problem.

We now turn our attention to showing that a strongly polite theory with respect to a set of sorts is shiny with respect to that set, assuming that the theory has a decidable quantifier-free satisfiability problem. The result holds for any set of sorts S since the computation of the $\mathsf{minmods}_{\mathcal{T},S}$ function will not rely on enumerating interpretations. This circumvents the restriction that $S = \Sigma^S$ imposed in [RRZ05]. We begin by proving lemmas relating the $\mathsf{minmods}$ function with equivalence classes.

Lemma 1.6. *Let Y be a finite set of variables and $E \sqsubseteq Y^2$ a sort-wise family of equivalence relations over Y. Given the arrangement formula δ_E^Y and a sort $\sigma \in \tau(Y)$, the number of equivalence classes in the quotient set of Y_σ by E_σ is computable in at most $\mathcal{O}(|Y_\sigma|^2)$ steps.*

Proof. Consider Algorithm 1 to compute the number of equivalence classes in the quotient set of Y_σ by E_σ, denoted in the sequel by $\left|{}^{Y_\sigma}\!/_{E_\sigma}\right|$, given the arrangement formula δ_E^Y and sort σ.

Algorithm 1 – Counts equivalence classes given an arrangement formula and a sort

Input: an arrangement formula δ_E^Y, and a sort σ
Output: cardinality of the quotient set $\left|{}^{Y_\sigma}\!/_{E_\sigma}\right|$
1: Construct a graph G_σ with nodes Y_σ and edge set $\left\{(x,y) : (x \cong_\sigma y) \in \delta_{E_\sigma}^{Y_\sigma}\right\}$
2: Compute connected components of G_σ
3: Return number of connected components of the graph

In step 1, the adjacency list for G_σ is built from $\delta_{E_\sigma}^{Y_\sigma}$. This can be made in $\mathcal{O}(|Y_\sigma|^2)$. To compute the number of connected components of the graph, we refer to the algorithm described by Hopcroft and Tarjan in [HT73] which has time complexity of the order of

$$\max\{|Ed|, |V|\} = \max\left\{\left|\left\{(x,y) : (x \cong_\sigma y) \in \delta_{E_\sigma}^{Y_\sigma}\right\}\right|, |Y_\sigma|\right\} \leq |Y_\sigma|^2 \ .$$

So the time complexity of Algorithm 1 is $\mathcal{O}(|Y_\sigma|^2)$. □

We now show that a strongly polite theory with respect to a set of sorts is stably finite with respect to that set.

Lemma 1.7. *A strongly polite theory with respect to a set of sorts S is stably finite with respect to that set.*

Proof. Let φ be a quantifier-free formula satisfied by a model $\mathcal{A}$ and let FS be the subset of sorts of S for which A is finite. Then $\mathcal{A}$ satisfies the formula $\exists \vec{w}\ \mathsf{s\text{-}witness}(\varphi)$ by the first condition of strong finite witnessability. Let $\mathcal{A}'$ be a model equivalent modulo $\vec{w}$ to $\mathcal{A}$ satisfying $\mathsf{s\text{-}witness}(\varphi)$ and consider the family of equivalence relations of variables of $\mathsf{s\text{-}witness}(\varphi)$ induced by $\mathcal{A}'$, i.e.,

$$E_{\mathcal{A}'} = \bigcup_{\sigma \in \mathsf{FS}} \{(x,y) : x^{\mathcal{A}'} = y^{\mathcal{A}'} \text{ and } x, y \in \mathsf{vars}_\sigma(\mathsf{s\text{-}witness}(\varphi))\} \ .$$

Take $Y = \bigcup_{\sigma \in \mathsf{FS}} \mathsf{vars}_\sigma(\mathsf{s\text{-}witness}(\varphi))$. Then $\mathcal{A}'$ satisfies $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_{E_{\mathcal{A}'}}^Y$. Observe that $\left|{}^{Y_\sigma}\!/_{E_{\mathcal{A}'\sigma}}\right| \leq |A'_\sigma| = |A_\sigma|$ for all $\sigma \in \mathsf{FS}$. By the second property of strong finite witnessability, we obtain that there is a $\mathcal{T}$-model $\mathcal{B}$ such that

$$B_\sigma = \left[\!\!\left[\mathsf{vars}_\sigma\left(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_{E_{\mathcal{A}'}}^Y\right)\right]\!\!\right]^{\mathcal{B}}, \text{ for all } \sigma \in S \ .$$

From this, we obtain that all $|B_\sigma|$ are finite for $\sigma \in S$, and that $|B_\sigma| = \left|Y_\sigma / E_{\mathcal{A}'\sigma}\right|$ for all $\sigma \in \mathsf{FS}$. So $|B_\sigma| \leq |A_\sigma|$ for all $\sigma \in \mathsf{FS}$ as desired. □

All the following lemmas assume that $\mathcal{T}$ is a strongly polite theory with respect to a set of sorts S and that φ is a quantifier-free satisfiable formula.

Lemma 1.8. *Let* $Y = \mathsf{vars}(\mathsf{s\text{-}witness}(\varphi))$ *and* $E \sqsubseteq Y^2$. *If* $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ *is* $\mathcal{T}$*-satisfiable, then* $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ *has a model with* $\left|Y_\sigma / E_\sigma\right|$ *as the cardinality of the* σ*-domain, for each* $\sigma \in S$.

Proof. By the second property of strong finite witnessability, since $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ is $\mathcal{T}$-satisfiable, there exists a $\mathcal{T}$-model $\mathcal{A}$ such that

$$A_\sigma = [\![\mathsf{vars}_\sigma\left(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y\right)]\!]^{\mathcal{A}}, \text{ for all } \sigma \in S \ .$$

Since $Y = \mathsf{vars}(\mathsf{s\text{-}witness}(\varphi))$, the cardinality of A_σ is exactly $\left|Y_\sigma / E_\sigma\right|$ for each $\sigma \in S$. □

Lemma 1.9. *Let* $Y = \mathsf{vars}(\mathsf{s\text{-}witness}(\varphi))$. *For all* $\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ *there is an* $E \sqsubseteq Y^2$ *such that*

$$\mathsf{Sat}(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y) = 1$$

and

$$\left|Y_\sigma / E_\sigma\right| = \bar{k}[\sigma] \text{ for each } \sigma \in S \ .$$

Proof. Choose $\bar{k}$ and let $\mathcal{A}$ be a $\mathcal{T}$-model of φ such that $|A_\sigma| = \bar{k}[\sigma]$ for each $\sigma \in S$. We have that $\mathcal{A}$ satisfies $\exists \vec{w}\ \mathsf{s\text{-}witness}(\varphi)$ by the first property of strong finite witnessability. Take $\mathcal{A}'$ as a model that satisfies $\mathsf{s\text{-}witness}(\varphi)$ and that is equivalent modulo $\vec{w}$ to $\mathcal{A}$. Observe that by definition of equivalent modulo $\vec{w}$, $|A_\sigma| = |A'_\sigma|$.

Now, consider the equivalence relation induced by $\mathcal{A}'$, $E_{\mathcal{A}'}$,

$$E_{\mathcal{A}'} = \bigcup_{\sigma \in S} \{(x, y) : x^{\mathcal{A}'} = y^{\mathcal{A}'} \text{ and } x, y \in \mathsf{vars}_\sigma(\mathsf{s\text{-}witness}(\varphi))\} \ .$$

Clearly, $\mathcal{A}'$ satisfies $\delta_{E_{\mathcal{A}'}}^Y$ and so it satisfies $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_{E_{\mathcal{A}'}}^Y$. Observe that $\left|Y_\sigma / E_{\mathcal{A}'\sigma}\right| \leq |A'_\sigma| = |A_\sigma| = \bar{k}[\sigma]$. Hence, by Lemma 1.8, we obtain that there is a $\mathcal{T}$-model $\mathcal{B}$ of $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_{E_{\mathcal{A}'}}^Y$ such that $|B_\sigma| = \left|Y_\sigma / E_{\mathcal{A}'\sigma}\right| \leq |A'_\sigma| = \bar{k}[\sigma]$ for each $\sigma \in S$. So, since $\mathcal{B}$ satisfies φ and $\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ we can conclude that $|B_\sigma| = \bar{k}[\sigma]$ for each $\sigma \in S$. That is, $\left|Y_\sigma / E_{\mathcal{A}'\sigma}\right| = \bar{k}[\sigma]$ for each $\sigma \in S$. □

Proposition 1.3. *A strongly polite theory with respect to a set of sorts* S *with a decidable quantifier-free satisfiability problem is shiny with respect to* S. *Moreover, Algorithm 2 computes the* $\mathsf{minmods}_{\mathcal{T},S}$ *function.*

Proof. Let $\mathcal{T}$ be a strongly polite theory with respect to a set of sorts S. By definition of strong politeness, $\mathcal{T}$ is smooth with respect to S. Moreover, by Lemma 1.7, $\mathcal{T}$ is stably finite with respect to S and so, in order to show that $\mathcal{T}$ is shiny with respect to S, it remains to prove that $\mathsf{minmods}_{\mathcal{T},S}$ is computable. Consider Algorithm 2 that returns, given an input formula φ, a set denoted in the sequel by $\mathsf{MM\_alg}(\varphi)$. The algorithm starts by building a set of relevant tuples of S-cardinalities of models, and proceeds by finding the minimal elements of this set with respect to the product order defined as usual, i.e., $\bar{k} \leq \bar{k}'$ iff $\bar{k}[\sigma] \leq \bar{k}'[\sigma]$ for all $\sigma \in S$. The method used for finding the minimal elements in a poset is described in [DKM+11].

Algorithm 2 – MM_alg algorithm – computes the $\mathsf{minmods}_{\mathcal{T},S}$ function for a strongly polite theory $\mathcal{T}$ with respect to a set of sorts S

Input: φ, a $\mathcal{T}$-satisfiable quantifier-free formula
Output: $\mathsf{minmods}_{\mathcal{T},S}(\varphi)$
1: $Y = \mathsf{vars}_S(\mathsf{s\text{-}witness}(\varphi))$
2: $\mathsf{Card} = \left\{ \langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S} : E \sqsubseteq Y^2 \text{ and } \mathsf{Sat}(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y) = 1 \right\}$
3: $\mathsf{MM} = \emptyset$
4: **for** $\bar{k} \in \mathsf{Card}$ do
5: **if** $\neg \exists \bar{m} \in \mathsf{MM} : \bar{k}[\sigma] \geq \bar{m}[\sigma]$ for all $\sigma \in S$
6: **then** $\mathsf{MM} = \{\bar{k}\} \cup \mathsf{MM} \setminus \{\bar{k}' \in \mathsf{MM} : \bar{k}[\sigma] \leq \bar{k}'[\sigma] \text{ for all } \sigma \in S\}$
7: return MM

(1) MM_alg terminates:
It is enough to see that Y is a finite set and so also the set Card.

(2) $\mathsf{MM\_alg}(\varphi) \subseteq \mathsf{minmods}_{\mathcal{T},S}(\varphi)$:
By Lemma 1.8, we know that Card only contains tuples of S-cardinalities of models of φ. Let $E \sqsubseteq Y^2$ be such that $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ is satisfiable and $\langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S} \in \mathsf{MM\_alg}(\varphi)$. Suppose by contradiction that

$$\langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S} \notin \mathsf{minmods}_{\mathcal{T},S}(\varphi) \ .$$

Then, by Lemma 1.2, there is a $\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ such that $\bar{k} \neq \langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S}$ and $\bar{k}[\sigma] \leq \left| Y_\sigma / E_\sigma \right|$ for each $\sigma \in S$. By Lemma 1.9, let $E' \sqsubseteq Y^2$ be such that $\mathsf{Sat}(\mathsf{s\text{-}witness}(\varphi) \wedge \delta_{E'}^Y) = 1$ and $\left| Y_\sigma / {E'}_\sigma \right| = \bar{k}[\sigma]$ for each $\sigma \in S$. Then $\left| Y_\sigma / {E'}_\sigma \right| \leq \left| Y_\sigma / E_\sigma \right|$ for each $\sigma \in S$. Observe that $\langle \left| Y_\sigma / {E'}_\sigma \right| \rangle_{\sigma \in S} \in \mathsf{Card}$ at step 2. Then, either $\langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S}$ would be removed from MM at step 6 when $\langle \left| Y_\sigma / {E'}_\sigma \right| \rangle_{\sigma \in S}$ is added or it would have never been added to the MM set if $\langle \left| Y_\sigma / {E'}_\sigma \right| \rangle_{\sigma \in S}$ was already there, since it fails the condition at step 5. Both these cases contradict with $\langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S} \in \mathsf{MM\_alg}(\varphi)$ and thus, $\langle \left| Y_\sigma / E_\sigma \right| \rangle_{\sigma \in S} \in \mathsf{minmods}_{\mathsf{T},S}(\varphi)$.

(3) $\mathsf{minmods}_{\mathcal{T},S}(\varphi) \subseteq \mathsf{MM\_alg}(\varphi)$:
Let $\bar{k} \in \mathsf{minmods}_{\mathcal{T},S}(\varphi)$ and, by Lemma 1.9, let $E \sqsubseteq Y^2$ be such that $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_E^Y$ is satisfiable and $\left| Y_\sigma / E_\sigma \right| = \bar{k}[\sigma]$ for each $\sigma \in S$. Suppose

by contradiction that $\bar{k} \notin \mathsf{MM\_alg}(\varphi)$. Then, by definition of $\mathsf{MM\_alg}$, there is an $\langle \left|Y_\sigma/R_\sigma\right| \rangle_{\sigma \in S} \in \mathsf{MM}$ such that $\left|Y_\sigma/R_\sigma\right| \leq \bar{k}[\sigma] = \left|Y_\sigma/E_\sigma\right|$ for each σ. By Lemma 1.8, let $\mathcal{A}$ be the model of $\mathsf{s\text{-}witness}(\varphi) \wedge \delta_R^Y$ such that $|A_\sigma| = \left|Y_\sigma/R_\sigma\right|$ for each $\sigma \in S$. Then $\mathcal{A}$ is a model of φ and so, by definition of $\mathsf{minmods}$, the tuple $\bar{k} \notin \mathsf{minmods}_{\mathcal{T},S}(\varphi)$, which is a contradiction. So $\bar{k} \in \mathsf{MM\_alg}(\varphi)$.

So the proposed algorithm $\mathsf{MM\_alg}$ terminates and computes the $\mathsf{minmods}_{\mathcal{T},S}$ function. □

Combining Propositions 1.2 and 1.3 we obtain the equivalence between shininess and strong politeness in the many-sorted case when the theories are equipped with a quantifier-free satisfiability solver. It should be made clear that, in practice, the requirement that the theories have satisfiability solvers is not a significant restriction since shiny and strongly polite theories were proposed in view of a more general Nelson-Oppen result, which is about combination of satisfiability solvers.

Theorem 1.1. *Let $\mathcal{T}$ be a first-order Σ-theory with a decidable quantifier-free satisfiability problem and $S \subseteq \Sigma^S$ a set of sorts. Then, the following statements are equivalent:*

1. *$\mathcal{T}$ is shiny with respect to S;*
2. *$\mathcal{T}$ is strongly polite with respect to S.*

1.4 Combination method for many-sorted shiny theories

In the previous section we showed the equivalence between many-sorted shiny and strongly polite theories. Because of this and the fact that there exists a Nelson-Oppen combination method for many-sorted strongly polite theories, see [JB10a], we get a Nelson-Oppen combination theorem for many-sorted shiny theories as a consequence.

Furthermore, we obtain two combination methods – one based on the constructive translation to strongly polite theories and the other based on directly making use of the $\mathsf{minmods}_{\mathcal{T},S}$ function of shiny theories. The latter method, in the one-sorted case, will coincide with the combination method for shiny theories introduced in [TZ05].

Theorem 1.2 (Combination theorem for many-sorted shiny theories)**.** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be theories with decidable quantifier-free satisfiability problems with no function or predicate symbols in common, and denote their set of sorts in common by S. If $\mathcal{T}_2$ is shiny with respect to S, then $\mathcal{T}_1 \cup \mathcal{T}_2$ has a decidable quantifier-free satisfiability problem.*

This theorem is a simple consequence of the following proposition, Proposition 1.4, where the satisfiability procedure for $\mathcal{T}_1 \cup \mathcal{T}_2$ is explicitly constructed using the satisfiability procedures for each of the theories.

We recall from the proof of Proposition 1.2 that a shiny theory $\mathcal{T}$ with respect to a set of sorts S with a decidable quantifier-free satisfiability problem has a strong witness function $\mathsf{s\text{-}witness} : \mathsf{QF}(\Sigma) \to \mathsf{QF}(\Sigma)$ defined as follows

$$\mathsf{s\text{-}witness}(\varphi) = \varphi \wedge \Omega,$$

where Ω is

$$\bigwedge_{\substack{E \sqsubseteq \mathsf{vars}(\varphi)^2 \\ \mathsf{Sat}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)=1}} \left(\delta_E^{\mathsf{vars}(\varphi)} \to \bigvee_{\bar{k} \in \mathsf{MM}(\varphi,E)} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]} \right) \ ,$$

where $\mathsf{MM}(\varphi, E)$ is the set $\mathsf{minmods}_{\mathcal{T},S}\left(\varphi \wedge \delta_E^{\mathsf{vars}(\varphi)}\right)$ and $\gamma_{\bar{k}[\sigma]}$ is

$$\begin{cases} \bigwedge_{\substack{i,j=1 \\ i \neq j}}^{\bar{k}[\sigma]} \neg(w_{i,\sigma} \cong_\sigma w_{j,\sigma}), & \text{if } \bar{k}[\sigma] > 1 \\ (w_{1,\sigma} \cong_\sigma w_{1,\sigma}), & \text{if } \bar{k}[\sigma] = 1 \end{cases}$$

with $w_{1,\sigma}, \ldots, w_{\bar{k}[\sigma],\sigma}$ being distinct fresh σ-variables.

Proposition 1.4. *Let $\mathcal{T}_i$ be Σ_i-theories with a decidable quantifier-free satisfiability problem, for $i = 1, 2$, such that $\Sigma_1^P \cap \Sigma_2^P = \emptyset$ and $\Sigma_1^F \cap \Sigma_2^F = \emptyset$. Assume that $\mathcal{T}_2$ is shiny with respect to $S = \Sigma_1^S \cap \Sigma_2^S$. Then for every conjunction Γ_1 of Σ_1-literals and Γ_2 of Σ_2-literals the following are equivalent:*

1. *$\Gamma_1 \wedge \Gamma_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$-satisfiable;*
2. *there exists $E \sqsubseteq Y^2$, where Y is $\mathsf{vars}_S(\mathsf{s\text{-}witness}_{\mathcal{T}_2,S}(\Gamma_2))$, such that*

$$\begin{aligned} &\mathcal{T}_1 \models \Gamma_1 \wedge \delta_E^Y \text{ and} \\ &\mathcal{T}_2 \models \mathsf{s\text{-}witness}_{\mathcal{T}_2,S}(\Gamma_2) \wedge \delta_E^Y \ . \end{aligned}$$

3. *there exists $E \sqsubseteq Y^2$, where Y is $\mathsf{vars}_S(\Gamma_1) \cap \mathsf{vars}_S(\Gamma_2)$, such that*

$$\begin{aligned} &\mathcal{T}_1 \models \Gamma_1 \wedge \delta_E^Y \wedge \left(\bigvee_{\bar{k} \in \mathsf{MM}} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]} \right) \text{ and} \\ &\mathcal{T}_2 \models \Gamma_2 \wedge \delta_E^Y \ , \end{aligned}$$

where MM is $\mathsf{minmods}_{\mathcal{T}_2,S}(\Gamma_2 \wedge \delta_E^Y)$.

Proof. The equivalence between 1. and 2. follows from the combination proposition, Proposition 2, in [JB10a], capitalizing on the equivalence of many-sorted shiny and strongly polite theories established in Theorem 1.1.

We now show 1.→3. and 3.→1. separately. Suppose $\Gamma_1 \wedge \Gamma_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$-satisfiable. Then there is a $\mathcal{T}_1 \cup \mathcal{T}_2$-model $\mathcal{A}$ that satisfies $\Gamma_1 \wedge \Gamma_2$. Furthermore, let $Y = \mathsf{vars}_S(\Gamma_1) \cap \mathsf{vars}_S(\Gamma_2)$ and δ_E^Y be the arrangement formula induced by

$$E_{\mathcal{A}} = \bigcup_{\sigma \in S} \{(x, y) : x^{\mathcal{A}} = y^{\mathcal{A}} \text{ and } x, y \in Y_\sigma\} \ .$$

Obviously $\mathcal{A}$ is a $\mathcal{T}_1$-model of $\Gamma_1 \wedge \delta_E^Y$ and a $\mathcal{T}_2$-model of $\Gamma_2 \wedge \delta_E^Y$. Finally, since $\mathcal{A}$ is a model of $\Gamma_2 \wedge \delta_E^Y$, by Lemma 1.2, the cardinalities of the domains A_σ for sorts $\sigma \in S$ must be larger or equal than $\bar{k}[\sigma]$ for some $\bar{k} \in \mathsf{minmods}_{\mathcal{T}_2,S}(\Gamma_2 \wedge \delta_E^Y)$. Hence, $\mathcal{A} \models (\bigvee_{\bar{k} \in \mathsf{MM}} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]})$.

For the other direction, suppose we have a $\mathcal{T}_1$-model $\mathcal{A}_1$ satisfying $\Gamma_1 \wedge \delta_E^Y \wedge (\bigvee_{\bar{k} \in \mathsf{MM}} \bigwedge_{\sigma \in S} \gamma_{\bar{k}[\sigma]})$ and a $\mathcal{T}_2$-model $\mathcal{A}_2$ satisfying $\Gamma_2 \wedge \delta_E^Y$. Take the tuple $\bar{K}$ of cardinalities of domains of $\mathcal{A}_1$ with sorts in S. Clearly, there is a $\bar{k} \in \mathsf{minmods}_{\mathcal{T}_2,S}(\Gamma_2 \wedge \delta_E^Y)$ such that $\bar{k}[\sigma] \leq \bar{K}[\sigma]$ for all $\sigma \in S$. By smoothness of $\mathcal{T}_2$, we know there is a $\mathcal{T}_2$-model $\mathcal{B}'$ of $\Gamma_2 \wedge \delta_E^Y$ such that $|B'_\sigma| = \bar{K}[\sigma]$ for all $\sigma \in S$. Since both $\mathcal{A}_1$ and $\mathcal{B}'$ satisfy δ_E^Y and have the same cardinalities of domains for each $\sigma \in S$, we can construct a $\mathcal{T}_1 \cup \mathcal{T}_2$-model of $\Gamma_1 \wedge \Gamma_2$ via Theorem 2.5 in the extended version, see [JB10b], of [JB10a]. □

This result provides a way to effectively construct the satisfiability procedure for $\mathcal{T}_1 \cup \mathcal{T}_2$, and so Theorem 1.2, which states the existence of such procedure, follows immediately.

1.5 Conclusion and Future Work

In this chapter we proved a Nelson-Oppen theorem for the combination of a many-sorted shiny theory with an arbitrary theory, extending to the many-sorted case the work in [TZ05]. For this, we investigated the relationship between the notions of shininess and strong politeness in the many-sorted case. We showed that, in the many-sorted case, a shiny theory with respect to a set of sorts (and equipped with a decidable quantifier-free satisfiability problem) is strongly polite with respect to the same set. On the other hand we were also able to prove that a strongly polite theory with respect to a set of sorts (with a decidable quantifier-free satisfiability problem) is shiny with respect to the same set. These results show that the classes of shiny and strongly polite theories with a decidable quantifier-free satisfiability problem are, in fact, the same.

In the future, we intend to investigate more general conditions that a theory should satisfy in order to be combined with an arbitrary theory by a Nelson-Oppen

method. More concretely, we leave as future work the investigation of a class of theories strictly containing the shiny/strongly polite theories for which there exists an indiscriminate Nelson-Oppen method, in the sense that they can be combined with an arbitrary theory with a decidable quantifier-free satisfiability problem.

Chapter 2

Generalized Probabilistic Satisfiability

2.1 Introduction

For many years, the satisfiability problem for propositional logic (SAT) has been extensively studied both for theoretical purposes, such as complexity theory, and for practical purposes. In spite of its NP-completeness [Coo71], 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 [BHvM09].

Naturally, people started extending this problem to more expressive frameworks: for instance in Satisfiability Modulo Theories [DMB11], instead of working in propositional logic, one can try to decide if a formula is valid in some specific first-order theory. As we have seen in the previous chapter, this area is also very fertile both in fundamental aspects such as satisfiability procedure combination, and practical aspects of the implementations of the solvers. One other direction is to extend propositional logic with probabilities. The probabilistic satisfiability problem (PSAT) was originally formulated by George Boole [Boo53] and later by Nilsson [Nil86]. 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 [GKP88, FB11, CI13, BCF15, FB15].

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, and has applications in the analysis of the security of cryptographic protocols and on estimating the probability of existence of attacks [MC17]. 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 [FHM90], 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 [CI13, BCF15].

Mixed-Integer Programming (MIP) [PS82] 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 [CH99] in order to reduce GenPSAT problems to suitable MIP problems.

As observed in many NP-complete problems [CKT91], GenPSAT also presents a phase transition behaviour. By solving batches of parametrized random GenPSAT problems, 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 [CKT91].

As the main contribution of this chapter, 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 2.2 we briefly recall the PSAT problem; in Section 2.3 we carefully define the GenPSAT problem and establish some results on its complexity; Section 2.4 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 2.5 we analyse the presence of phase transition; finally, in Section 2.6, we assess our contributions and discuss future work.

2.2 Preliminaries

Let us begin by fixing a set of propositional variables $\mathcal{P} = \{x_1, \ldots, x_n\}$. We define the set of *classical propositional formulas* as

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

Observe that the other logical connectives $\rightarrow, \vee, \leftrightarrow$ can be defined by abbreviation, as usual. A *literal* is either a propositional variable or its negation. A *propositional clause* is a non-empty disjunction of one or more literals.

A *propositional valuation* is a map $v : \mathcal{P} \rightarrow \{0, 1\}$, which is extended to

propositional formulas as usual. We say that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$. This notion is extended to sets of propositional formulas as usual. Let $\mathcal{V}^* = \{v_1, \ldots, v_{2^n}\}$ be the set of all valuations defined over variables of $\mathcal{P}$. We define a *probability distribution* π over $\mathcal{V}^*$ as a probability vector of size 2^n.

A *simple probabilistic formula* is an expression of the form $\mathsf{Pr}(c) \boxtimes p$, where c is a clause, $p \in \mathbb{Q}$, $0 \leq p \leq 1$ and $\boxtimes \in \{=, \leq, \geq\}$. We say that a probability distribution π *satisfies* a formula $\mathsf{Pr}(c) \boxtimes p$ if

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

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

We now recall the PSAT problem [Nil86, GKP88, FB11].

Definition 2.1 (PSAT problem)**.** *Given a set of propositional variables $\mathcal{P}$ and a set of simple probabilistic formulas $\Sigma = \{\mathsf{Pr}(c_i) \boxtimes p_i \mid 1 \leq i \leq k\}$, 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 $\{\mathsf{Pr}(c_i) \boxtimes_i p_i \mid 1 \leq i \leq k\}$ can be formulated algebraically as the problem of finding a solution π for the system of inequalities

$$\begin{cases} V\pi \boxtimes p \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} ,$$

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, $p = [p_i]$ is the k vector of all p_i and $\boxtimes = [\boxtimes_i]$ is the k vector of all $\boxtimes_i$.

The SAT problem can be modelled as a PSAT instance where the entries p_i of the probability vector are all identical to 1. The PSAT problem was shown to be NP-complete [GKP88, FHM90], even when the clauses consist of the disjunction of only two literals, 2-PSAT.

2.3 The 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

$$\sum_{i=1}^{\ell}(a_i\mathsf{Pr}(\varphi_i)) \bowtie p \ ,$$

where $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$, $\bowtie \in \{\geq, <, \neq\}$, $\ell \in \mathbb{N}$ and $a_i, p \in \mathbb{Q}$. Observe that formulas with the relational symbols $\leq, >$ can be obtained by abbreviation and formulas with $=$ are obtained as a combination of probabilistic formulas. An *atomic probabilistic formula* is a probabilistic formula where each φ_i is a propositional variable. We say that a probability distribution π *satisfies* a formula $\sum_{i=1}^{\ell}(a_i\mathsf{Pr}(\varphi_i)) \bowtie p$ if

$$\sum_{i=1}^{\ell}\left(a_i\left(\sum_{j=1}^{2^n} v_j(\varphi_i)\cdot\pi_j\right)\right) \bowtie p \ .$$

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

An *instance* of GenPSAT is a pair (Γ, Σ) where Γ is a set of propositional formulas (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 (Γ, Σ) if it satisfies the set of probabilistic formulas

$$\Xi_{(\Gamma,\Sigma)} = \Sigma \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Gamma\} \ . \tag{2.1}$$

Definition 2.2 (GenPSAT problem)**.** *Given a* GenPSAT *instance* (Γ, Σ)*, the Generalized Probabilistic Satisfiability problem (*GenPSAT*) consists in determining whether there exists a probability distribution* π *over* $\mathcal{V}^*$ *that satisfies* (Γ, 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 formulas 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 [Ros09].

Example 2.1. *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 variables:* 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:*

$$\Gamma_1 = \left\{ \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (P_i \wedge \neg P_j \wedge \neg P_k), \quad \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (X_i \wedge \neg X_j \wedge \neg X_k), \quad \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (H_i \wedge \neg H_j \wedge \neg H_k) \right\} \ .$$

Furthermore, the host cannot open neither the chosen door nor the door with the prize and so we also impose the following constraints:

$$\Gamma_2 = \bigcup_{i\in\{1,2,3\}} \{P_i \to \neg H_i,\, X_i \to \neg H_i\} \ .$$

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 *and* winning by keeping*, respectively, as*

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

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*

$$(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) < \mathsf{Pr}(\mathsf{WbK})\})\,, \ \textit{where}\ \Gamma = \Gamma_1 \cup \Gamma_2\ .$$

As expected, this instance is not satisfiable and the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})\})$ *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* $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \neq \frac{2}{3}\})$ *is unsatisfiable and that the instance* $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) = \frac{2}{3}\})$ *is satisfiable. All these instances were checked using the tool we implemented, [CMC16a].* ◊

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

Given a GenPSAT instance (Γ, Σ), where Γ contains m formulas and Σ is composed of k probabilistic formulas, we follow the lines of Nilsson [Nil86] 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 $\sum_{u=1}^{\ell} a_u^i \mathsf{Pr}(\varphi_u^i) \bowtie_i p_i$ of $\Xi_{(\Gamma,\Sigma)}$ as follows:

$$V_{ij} = \textstyle\sum_{u=1}^{\ell} a_u^i \cdot v_j(\varphi_u^i)\ .$$

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

$$\begin{cases} V\pi \bowtie p \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} . \tag{2.2}$$

Given a set of probabilistic formulas $\Omega = \left\{ \sum_{u=1}^{\ell} a_u^i \cdot v_j(\varphi_u^i) \bowtie_i p_i \mid 1 \leq i \leq k \right\}$ 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

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

and

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

Then, we can rewrite system (2.2) using the $[\Xi_{(\Gamma,\Sigma)}, \mathcal{V}^*]$-associated matrix V as

$$\begin{cases} V\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{2.3}$$

We now show that this problem is NP-complete. For this purpose, we first present the following lemma.

Lemma 2.1 ([FHM90, Chv83])**.** *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.*

Theorem 2.1 ([FHM90])**.** GenPSAT *is* NP*-complete.*

Proof. We begin by showing that GenPSAT is in NP by providing a polynomial sized certificate. Notice that Lemma 3.1 can be extended to rational coefficients simply by normalizing with the greatest denominator. Applying this result to the system (2.3) we conclude that there is a $(k+m+1) \times (k+m+1)$ matrix W, composed of columns of V, whose system

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{2.4}$$

has a solution iff the original system (2.3) has a solution. Furthermore, the obtained solutions from (2.4) can be mapped to solutions of (2.3) by inserting zeros in the appropriate positions. Since the solution of this system has $k+m+1$ elements, it constitutes the NP-certificate for the GenPSAT problem.

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

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

Lemma 2.2. *Given a* GenPSAT *instance* (Γ, Σ) *there exists an instance* (Γ', Σ') *in normal form such that* (Γ, Σ) *is satisfiable iff* (Γ', Σ') *is satisfiable. Moreover,* (Γ', Σ') *is obtained from* (Γ, Σ) *in polynomial time.*

Proof. Let (Γ, 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 $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(\varphi_i) \bowtie p$ be a formula in Σ and consider the atomic probabilistic formula obtained by replacing (when needed) each formula φ_i by a fresh variable y_i,

$$\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p \ .$$

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

We are left with the transformation of the formula

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

into 3-CNF using Tseitin's transformation [Tse68], which can increase linearly the size of the formula and add new variables to $\mathcal{P}$. The final Γ' is the set of conjuncts of the obtained 3-CNF formula. Since Tseitin's transformation preserves satisfiability of formulas, (Γ, Σ) is satisfiable iff (Γ', Σ') is satisfiable. □

2.4 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 [CH99]). With this, we present a reduction of GenPSAT to Mixed-Integer Programming (MIP), similarly to what was done for PSAT [CI13] and GPSAT [BCF15]. 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 [PS82]. Observe that this translation to MIP also serves as a proof that GenPSAT is in NP.

2.4.1 Linear Algebraic Formulation for GenPSAT

Lemma 2.3. *A* GenPSAT *instance in normal form* (Γ, 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:*

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

(2) $\mathcal{V}_0$ *satisfies* Γ*,*

(3) considering $p = [p_1, \ldots, p_k, 1]$ *and* $\bowtie = [\bowtie_1, \ldots, \bowtie_k, =]$*, the system*

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{2.5}$$

is satisfiable.

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

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

has a solution. And so, using Lemma 2.1, 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 Γ, 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 2.1 once more, we conclude that exists a $(k+1) \times k'$ matrix W, with $k' \leq k+1$, whose system has a positive solution ρ^*. The solution π for (2.5) is obtained from ρ^* 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 *(1), (2), (3)*, and let π denote the solution for (2.5). We are looking for a probability distribution π^* satisfying (Γ, Σ). For this purpose, let $\mathcal{V}_0 = \{v_{j_1}, \ldots, v_{j_{k'}}\} \subseteq \mathcal{V}$ denote the set of valuations underlying W according to condition *(2)*, and define $\pi^* = [\pi^*_i]$, where

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

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

- given $\gamma \in \Gamma$, we check that π^* verifies $\mathsf{Pr}(\gamma) = 1$ by observing that the last equality represented on W on (2.5) leads to $\sum_{s=1}^{k'} \pi_{j_s} = 1$ and so,

$$\begin{aligned}\sum_{j=1}^{2^n} v_j(\gamma) \cdot \pi_j^* &= \sum_{\{j|v_j(\gamma)=1\}} \pi_j^* \\ &= \sum_{s=1}^{k'} \pi_{j_s} \\ &= 1 \ .\end{aligned}$$

- given an atomic probabilistic formula $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ in Σ, we recall the definition of π^* and that π is a solution for (2.5) to conclude that

$$\begin{aligned}\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(y_i) \cdot \pi_j^* \right) &= \sum_{i=1}^{\ell} a_i \left(\sum_{s=1}^{k'} v_{j_s}(y_i) \cdot \pi_{j_s} \right) \\ &= \sum_{s=1}^{k'} \left(\sum_{i=1}^{\ell} a_i \cdot v_{j_s}(y_i) \right) \pi_{j_s} \bowtie p \ ,\end{aligned}$$

  i.e., π^* satisfies the formulas in Σ.

□

2.4.2 Translation to MIP

Regarding Lemma 2.3, given a GenPSAT instance (Γ, Σ) in normal form, with $|\Sigma| = k$ and $|\Gamma| = m$, 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 *(1), (2), (3)*, 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 variable 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 $\alpha_1, \ldots, \alpha_n$ represent the probability of the propositional variables $x_1, \ldots, x_n$, respectively, and following the reasoning of [CI13, BCF15] we model the non-linear constraint $\sum_{j=1}^{k'} h_{ij} \cdot \pi_j = \alpha_i$ as a linear inequality

$$\sum_{j=1}^{k'} b_{ij} = \alpha_i \ , \tag{val1}$$

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.,

$$0 \leq b_{ij} \leq h_{ij} \text{ and } h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j \ . \tag{val2}$$

We ensure that π represents a probability distribution by imposing that

$$\sum_{j=1}^{k'} \pi_j = 1 \ . \tag{sums1}$$

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

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

Notice that, if we have a total of m clauses in Γ, 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 $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ 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 $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \geq p$, we generate the linear inequality

$$\sum_{i=1}^{\ell} a_i \cdot \alpha_i \geq p \ . \tag{prob$_{\geq}$}$$

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,

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

and further introduce the linear constraint

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

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

$$\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) - p \neq 0, \tag{2.6}$$

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

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

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 $C = 1 + |p| + \sum_{i=1}^{\ell} |a_i|$, the inequality (2.6) holds if and only if there exists a fresh binary variable z_φ such that the following two strict inequalities hold simultaneously:

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

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

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

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 2.1. *The* GenToMIP *procedure transforms a* GenPSAT *instance in normal form* (Γ, Σ) *into a* MIP *problem whose size is polynomial on the size of* (Γ, 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 *(1), (2), (3)* of Lemma 2.3 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, 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.

Proposition 2.2. *A* GenPSAT *instance in normal form* (Γ, Σ) *is satisfiable iff Algorithm 3 returns* Sat.

Proof. Let (Γ, Σ) be a satisfiable GenPSAT instance in normal form, and also $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ and $\rho = [\rho_i]$ represent a set of valuations and a probability distribution given by Lemma 2.3 which satisfy conditions *(1), (2), (3)*. Then, consider the following values and afterwards let us check that they constitute an optimal solution for the MIP problem constructed at Algorithm 3: for each $1 \leq i \leq n$ and $1 \leq j \leq k'$, let

$$\begin{aligned} h^*_{ij} &= v_j(x_i), \\ b^*_{ij} &= h^*_{ij} \cdot \rho_j, \\ \pi^*_j &= \rho_j, \end{aligned}$$

$$\begin{aligned} \alpha^*_i &= \sum_{\{j | v_j(x_i) = 1\}} \rho_j, \\ \varepsilon^* &= \mathsf{min}\ \Delta, \end{aligned}$$

Algorithm 3 GenPSAT solver based on MIP

1: **procedure** GenPSAT(props $\{x_i\}_{i=1}^n$, form Γ, probform Σ)
2: **declare**: binary variables: h_{ij} for $i \in \{1,\ldots,n\}$, $j \in \{1,\ldots,k'\}$
3: $[0,1]$-variables: α_i, π_j, b_{ij} for $i \in \{1,\ldots,n\}$, $j \in \{1,\ldots,k'\}$
4: real variable: ε
5: **for** $j = 1$ to k' **do**
6: **for** each $(\bigvee_r x_r) \vee (\bigvee_s \neg x_s)$ in Γ **do**
7: add_const$(\sum_r h_{rj} + \sum_s (1 - h_{sj}) \geq 1)$ ▷ (gamma)
8: **for** $i = 1$ to n **do**
9: add_const$(\sum_j b_{ij} = \alpha_i)$ ▷ (val1)
10: **for** $j = 1$ to k' **do**
11: add_const$(0 \leq b_{ij} \leq h_{ij})$ ▷ (val2)
12: add_const$(h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j)$ ▷ (val2)
13: $aux \leftarrow 0$
14: **for** each $\sum a_i \cdot \mathsf{Pr}(x_i) \bowtie q$ in Σ **do**
15: **switch**$(\bowtie)$
16: **case** " $\geq$ " :
17: add_const$(\sum a_i \cdot \alpha_i \geq q)$ ▷ (prob$_\geq$)
18: **case** " $<$ " :
19: $aux \leftarrow 1$
20: set_obj(max ε) ▷ (obj)
21: add_const$(\sum a_i \cdot \alpha_i + \varepsilon \leq q)$ ▷ (prob$_<$)
22: **case** " $\neq$ " :
23: $aux \leftarrow 1$
24: $z \leftarrow$ fresh() ▷ z is a fresh binary variable
25: $C \leftarrow 1 + |q| + \sum |a_i|$
26: set_obj(max ε) ▷ (obj)
27: add_const$(\sum a_i \cdot \alpha_i - C \cdot z - \varepsilon \geq q - C)$ ▷ (prob$_\neq$)
28: add_const$(\sum a_i \cdot \alpha_i - C \cdot z + \varepsilon \leq q)$ ▷ (prob$_\neq$)
29: add_const$(\sum \pi_i = 1)$ ▷ (sums1)
30: **if** mip_sat() **then**
31: **if** $(aux == 0)$ or $(aux == 1$ and mip_objvalue() $> 0)$ **then**
32: **return** Sat
33: **return** Unsat

where $\Delta = \{q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid (\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) < q) \in \Sigma\} \cup$
$\cup \{C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\} \cup$
$\cup \{C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\}$,
and, for each atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q$,

$$z_\varphi^* = \begin{cases} 0, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha_i^* < q \\ 1, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha_i^* > q \end{cases} .$$

Now let us check that each linear constraint introduced into the MIP problem at Algorithm 3 is satisfied.

(gamma) $\{h_{ij}^*\}$ satisfy the constraints modelling Γ since each $v \in V_0$ satisfies Γ.

(val1) By definition of $\{b_{ij}^*\}$ and $\{h_{ij}^*\}$, we actually have

$$\sum_{j=1}^{k'} b_{ij}^* = \sum_{j=1}^{k'} h_{ij}^* \cdot \rho_j = \sum_{j=1}^{k'} v_j(x_i) \cdot \rho_j = \sum_{\{j \mid v_j(x_i)=1\}} \rho_j = \alpha_i^* .$$

(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 $\sum_{j=1}^{k'} \pi_j^* = 1$.

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

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

(prob$_\geq$) Let $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \geq q) \in \Sigma$ and notice that, since ρ satisfies conditions *(1), (2), (3)*, in particular it satisfies all the probabilistic formulas in Σ, and so $\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right) \geq q$, which implies that $\sum_{i=1}^{\ell} a_i \cdot \alpha_i^* \geq q$.

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

$$q - \sum_{i=1}^{\ell} (a_i \cdot \alpha_i^*) > 0. \tag{2.7}$$

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

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

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

$$C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) > 0 \tag{2.8}$$

and

$$C - C \cdot z_\varphi^* - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) > 0 \tag{2.9}$$

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

- if $q > \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*)$, then $z_\varphi^* = 0$ and it follows that
  $$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* - \varepsilon^* \geq \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) = q - C,$$
  and further,
  $$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) - C \cdot z_\varphi^* + \varepsilon^* \leq \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) + C \cdot z_\varphi^* + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) = q.$$

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

To finish the direct implication, notice that $\varepsilon^* > 0$ as a consequence of (2.7), (2.8) and (2.9), 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 returned Sat, and let us 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 $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ where, for each propositional variable $x_i \in \mathcal{P}$, $v_j(x_i) = h^*_{ij}$. Due to constraints (gamma) it is immediate to conclude that each valuation satisfies Γ. 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 2.3 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 2.2. *The* GenToMIP *algorithm is a correct polynomial time translation of* GenPSAT *to a* MIP *problem.*

2.5 Phase Transition

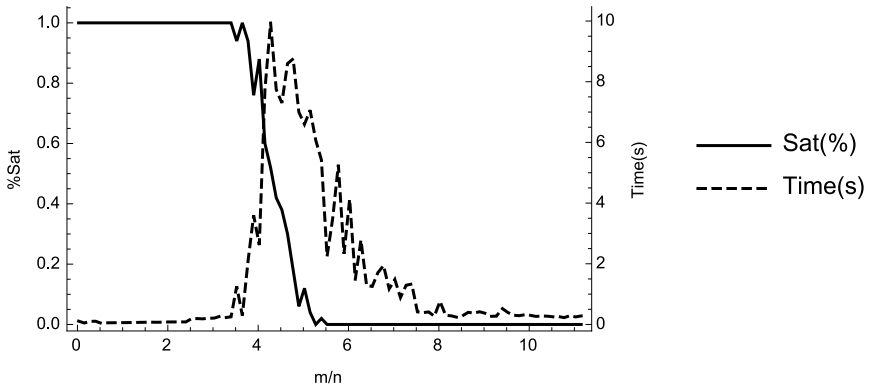
Phase transition is a phenomenon that marks a hardness shift in the solution of instances of a problem. This behaviour was observed in many NP-complete problems [CKT91], among which we highlight 3-SAT [GW94] and PSAT [FB11, FB15].

In this section, we study the GenPSAT phase transition, through an implementation of Algorithm 3 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 [GO15], 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 [CMC16a].

It was noted that, in random 3-SAT instances [GW94] 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

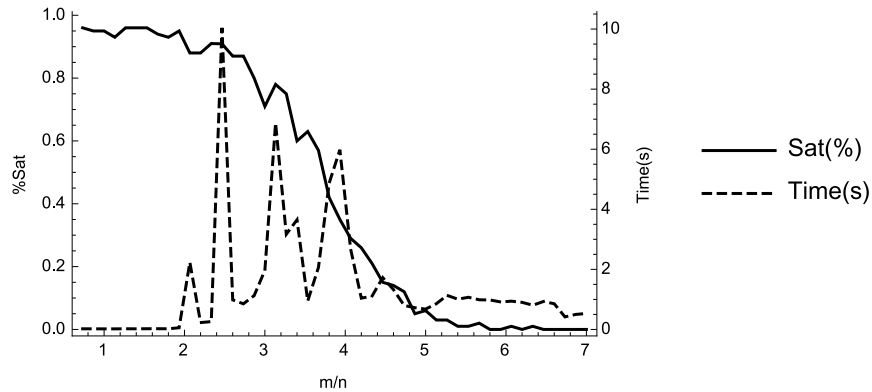
ratio values are almost certainly unsatisfiable and also easily solved. However, with values of the ratio very closed to this threshold, the instances are, on average, very hard to solve and there is no certainty on whether the problem is satisfiable or not. As we have already noted, any 3-SAT problem 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 [GW94], see Figure 2.1.

Figure 2.1 Phase transition for SAT seen as a GenPSAT instance, with $n = 20$.



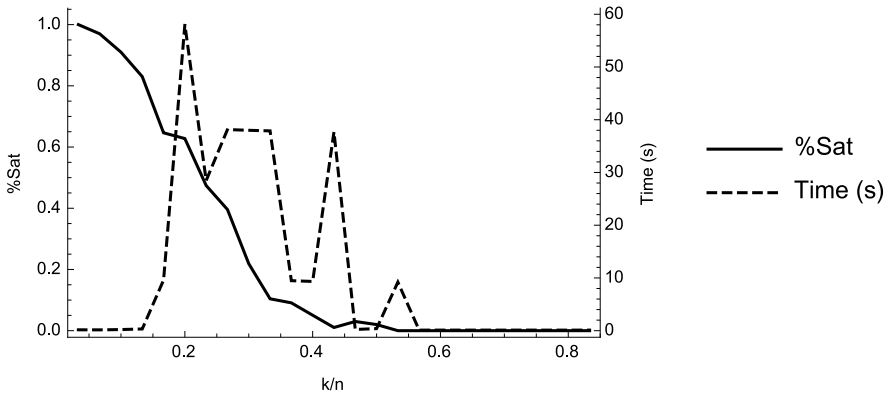
A deeper analysis of the probabilistic satisfiability problem PSAT [FB11, FB15] has shown the presence of a phase transition behaviour for PSAT for a ratio m/n, where m is the number of clauses and n is the number of variables. We tested random PSAT instances with the number of probabilistic formulas $k = 2$, $n = 15$ 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 2.2.

Figure 2.2 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 2.3.

Figure 2.3 PSAT phase transition seen as a GenPSAT instance, with $n = 30$ and $m = 40$.



In [BCF15], this phase transition analysis was performed on a generalization of the probabilistic satisfiability problem, GPSAT, which consists in Boolean combinations of simple probabilistic formulas.

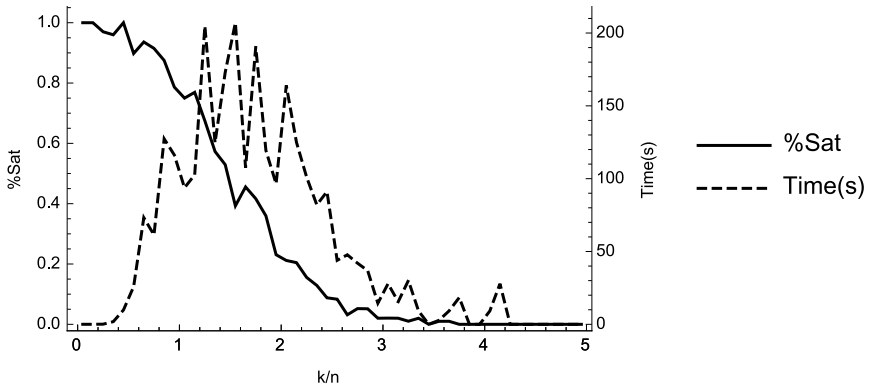
In what concerns our generalized version of probabilistic satisfiability GenPSAT, notice that a randomly sampled probabilistic formula can easily be inconsistent by itself, e.g., when it implies one of the probabilities is greater than 1. Because of this, the sampling of the coefficients was performed in such a way that this case does not occur. For example, to randomly sample formulas of the type $\sum a_i \mathsf{Pr}(\varphi_i) \geq p$, we randomly sample the a_i coefficients in $[0, 1]$, and the independent term p is sampled from the interval $[0, \sum_i a_i]$.

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 analyse 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

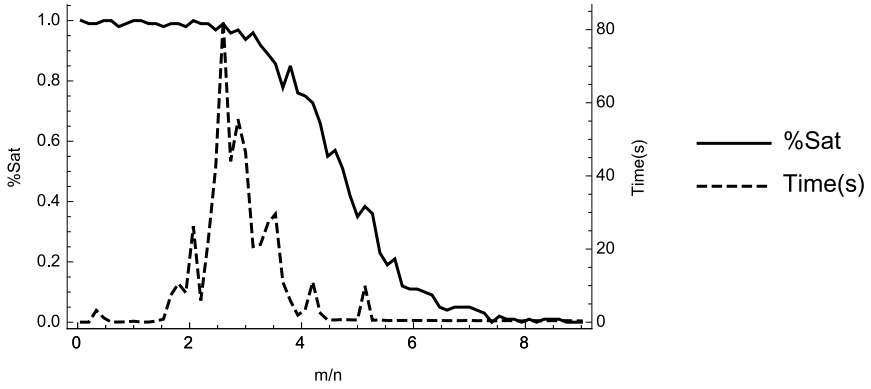
since a GenPSAT instance is more likely to be unsatisfiable. Figure 2.4 represents the phase transition for random GenPSAT instances with $n = 20$, $m = 10$ and k ranging from 1 to 100 in steps of 2. We generated 100 instances for each value of k.

Figure 2.4 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 2.5 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 2.5 Phase transition for random GenPSAT instances, with $n = 15$ and $k = 2$.



2.6 Conclusion and Future Work

Throughout this work 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 analysing the behaviour for different ratios. As future work, we leave open 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.

We built a tool that implements this algorithm, which although being able to solve reasonably sized instances, could always be improved and optimized. In this sense, we also performed preliminary tests on encoding the described reduction in SMT solvers Z3 and Yices. However, the obtained results were very similar to the ones presented within the chapter.

We leave as future work the study of the relationship between GenPSAT and *weighted* MaxSAT and the exploration of a solver via a reduction to this problem.

We also highlight how close the language of GenPSAT is to the language of the probabilistic logic of Fagin et al. As such, in the next chapter, we focus in generalizing the syntax to encompass the full language and develop a solver for the probabilistic logic of Fagin et al. This development opens doors to genuine applications of the probabilistic formalism in, for instance, information security.

Chapter 3

Classical Generalized Probabilistic Satisfiability

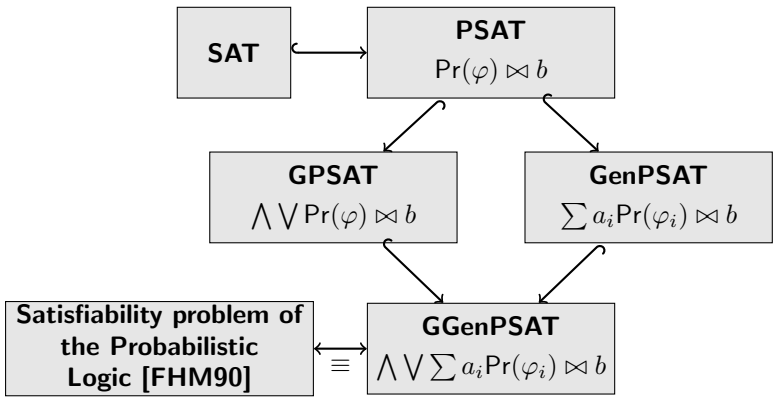
3.1 Introduction

The starting point of a deep analysis of the propositional satisfiability (SAT) problem was due to Cook in [Coo71], where it was shown that this problem is NP-complete. Given its simplicity and expressiveness, the SAT problem has become the standard NP-complete problem to study and, because of that, SAT solvers became extremely efficient. Due to this, several extensions and generalizations have been developed, taking advantage of the referred solvers. An example of this is the satisfiability modulo theories problem (SMT) [DMB11] where instead of working in propositional logic, one tries to decide if a formula is valid in some specific first-order theory. This area has had a great impact in industry, especially in hardware and software verification. One other direction for generalization of propositional satisfiability consists in the introduction of probabilities into the classical reasoning, allowing one to express quantitative assertions about propositional formulas.

In this sense, there was an effort to extend propositional logic in order to handle probabilistic reasoning. Fagin et al. [FHM90] developed a widely used probabilistic logic and showed that its satisfiability problem is NP-complete. Recently, several satisfiability solvers were proposed for fragments of this probabilistic logic. Finger and Bona developed a PSAT solver [FB11, FB15] in the context of the probabilistic satisfiability problem (PSAT) [Boo53, Nil86], which consists in deciding the satisfiability of a set of assignments of probabilities to propositional formulas. Afterwards, the PSAT problem was generalized to handle Boolean combinations of assignments of probabilities to propositional formulas leading to GPSAT in [BCF15]. After that, in [CCM17b], we introduced the generalized probabilistic satisfiability problem (GenPSAT) which consists in deciding the satisfiability of linear inequalities involving probabilities of classical propositional formulas, pre-

sented in the previous chapter. Given this, it is only natural to think about the extension of this problem to Boolean combinations of probabilistic formulas. This is our goal for this chapter: extend the GenPSAT problem to allow Boolean combinations of probabilistic formulas as well as present a solver for this more expressive problem.

Figure 3.1 Inclusion diagram of several fragments of the probabilistic logic



In this chapter, we present the classical generalized probabilistic satisfiability problem GGenPSAT, which consists in deciding the satisfiability of Boolean combinations of linear inequalities involving probabilities of classical propositional formulas. This problem was proved to be NP-complete in [FHM90]. We stress that the formulas expressible in GGenPSAT are precisely the formulas in the probabilistic logic by Fagin et al. We develop an algorithm for the GGenPSAT problem by constructing a polynomial reduction to the quantifier-free theory of linear integer and real arithmetic (QF_LIRA) [Kin14, BFT16]. Furthermore, we provide an implementation of the algorithm and study its phase transition behaviour. This study is done experimentally, and we detect two different parameters for which there is a clear phase transition on the satisfiability of random formulas. Furthermore, in the last section of the chapter, we model and verify, using the developed solver, two examples in information security, one related to hardware verification under faulty gates and the other which studies the effectiveness of Boolean masking against side-channel attacks.

As the main contribution in this chapter, we develop the theoretical framework that allows the translation between GGenPSAT and SMT problems, which then allows the implementation of a provably correct solver for GGenPSAT. With the GGenPSAT solver in hands, we are able to detect and study the phase transition behaviour. We also provide two detailed examples of application in information security of this probabilistic formalism, as well as the actual implementation in the developed tool.

This chapter is outlined as follows: in Section 3.2 we recall some basic notions on probabilistic satisfiability; in Section 3.3, we introduce the GGenPSAT problem; in Section 3.4, we present the polynomial reduction to SMT and prove the correctness of the algorithm; in Section 3.5, we describe the implemented tool and study the phase transition behaviour of the GGenPSAT problem; in Section 3.6 we develop two applications in information security; finally, Section 3.7 concludes the chapter and discusses avenues for further research.

3.2 Preliminaries

Let us begin by fixing a set of propositional variables $\mathcal{P} = \{x_1, \ldots, x_n\}$. The set of *classical propositional formulas* is defined, as usual, by

$$\mathrm{L_{CPL}} ::= \mathcal{P} \mid \neg \mathrm{L_{CPL}} \mid \mathrm{L_{CPL}} \wedge \mathrm{L_{CPL}} \ .$$

A *propositional literal* is either a propositional variable or its negation. A *propositional clause* is a non-empty disjunction of one or more propositional literals. A *propositional valuation* is a map $v : \mathcal{P} \to \{0, 1\}$, which is extended to propositional formulas as usual. We say that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$. This notion is extended to sets of propositional formulas as usual. Let $\mathcal{V}^* = \{v_1, \ldots, v_{2^n}\}$ be the set of all valuations defined over variables of $\mathcal{P}$. We define a *probability distribution* π over $\mathcal{V}^*$ as a probability vector of size 2^n.

We recall from [FHM90] the set of *probabilistic atoms* (used herein to define probabilistic formulas) composed by linear inequalities of probabilities of propositional formulas with rational coefficients:

$$\mathsf{PAt} ::= \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L_{CPL}}) + \ldots + \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L_{CPL}}) \geq \mathbb{Q} \ .$$

The set of *probabilistic formulas* is defined as a Boolean combination of probabilistic atoms as follows:

$$\mathsf{Prob} ::= \mathsf{PAt} \mid \neg \mathsf{Prob} \mid \mathsf{Prob} \wedge \mathsf{Prob} \ .$$

Observe that the other relational symbols $\{<, >, \leq, =, \neq\}$ can be defined by abbreviation, as well as the logical connectives $\rightarrow, \vee, \leftrightarrow$.

To interpret probabilistic formulas, we consider a probability distribution π over $\mathcal{V}^*$. The satisfaction relation is inductively defined as:

- $\pi \Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q$ iff $\displaystyle\sum_{i=1}^{\ell} \left(q_i \left(\sum_{j=1}^{2^n} v_j(\varphi_i) \cdot \pi_j \right) \right) \geq q$;

- $\pi \Vdash \neg\delta$ iff $\pi \nVdash \delta$;
- $\pi \Vdash \delta_1 \wedge \delta_2$ iff $\pi \Vdash \delta_1$ and $\pi \Vdash \delta_2$,

where $\delta, \delta_1, \delta_2 \in \mathsf{Prob}$, $q, q_i \in \mathbb{Q}$ and $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$ for $i \in \{1, \ldots, \ell\}$. A probability distribution π *satisfies* $\delta \in \mathsf{Prob}$ if $\pi \Vdash \delta$ and satisfies a set of probabilistic formulas if it satisfies each one of them.

3.3 The GGenPSAT problem

We now aim to extend the GenPSAT problem, presented in the last chapter, in order to cope with Boolean combinations of probabilistic atoms.

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

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

Despite the similarities between a GenPSAT and a GGenPSAT instance, the latter allows more expressive probabilistic formulas by allowing Boolean combinations of probabilistic atoms.

Definition 3.1 (GGenPSAT problem)**.** *Given a* GGenPSAT *instance* (Γ, Ψ)*, the Classical Generalized Probabilistic Satisfiability problem* (GGenPSAT) *consists in determining whether there exists a probability distribution* π *over* $\mathcal{V}^*$ *that satisfies* (Γ, Ψ)*.*

GGenPSAT extends the scope of PSAT and GenPSAT by dealing with Boolean combinations of probabilistic formulas. In this way, we are not only able to assign values to probabilities of propositional variables or linear inequalities involving them, but also able to express powerful probabilistic assertions. For instance, we can easily model and reason about a framework where a variable x is either true or false with probability 1 but we do not know which is the case:

$$\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1 \ .$$

To notice the impact of this generalization on the available models of a formula, consider the following examples.

Example 3.1. *Let* x, y *be two propositional variables and consider the propositional formula* φ *defined by* $\varphi \triangleq x \oplus y$*. Then, observe that the probabilistic formula*

$$\mathsf{Pr}(\neg(\varphi \leftrightarrow x)) = \frac{1}{2} \tag{3.2}$$

is satisfiable and consider the following truth table

	x	y	φ
v_1	*0*	*0*	*0*
v_2	*0*	*1*	*1*
v_3	*1*	*0*	*1*
v_4	*1*	*1*	*0*

The two valuations that satisfy the formula $\neg(\varphi \leftrightarrow x)$ *are* v_2 *and* v_4*, and so any probability distribution* π *assigning to* v_2 *probability* a *and to* v_4 *probability* b *satisfies* (3.2) *iff* $a + b = \frac{1}{2}$.

To showcase the expressiveness abilities of this wider formulation of probabilistic satisfiability, consider the additional hypothesis that the propositional variable y *is either true or false with probability 1, i.e.,*

$$\mathsf{Pr}(y) = 1 \vee \mathsf{Pr}(y) = 0 \ .$$

With this additional restriction, formula (3.2) *is not satisfiable any more since there is no probability distribution assigning either probability 1 or 0 to* y *that also complies with* $a + b = \frac{1}{2}$. ◊

Example 3.2. *Consider a game of* Odds and Evens *where players* A *and* B *play* $x, y \in \mathbb{Z}_2$*, respectively, and player* A *wins iff* $x \oplus y = 0$*. We can easily study the effectiveness and existence of strategies for this game in* GGenPSAT*: the term* $\mathsf{Pr}(\neg(x \oplus y))$ *represents the probability that player* A *wins the game; with this, we can determine that there is a strategy in which player* B *wins sometimes,* $\mathsf{Pr}(x \oplus y) > 0$*, and that player* A *wins twice as much as player* B*,* $\mathsf{Pr}(\neg(x \oplus y)) \geq 2 \cdot \mathsf{Pr}(x \oplus y)$ *by checking the satisfiability of such formulas.*

However, if we additionally assume that player A *always plays* 0*,* $\mathsf{Pr}(x) = 0$*, and that player* B *always plays the same,* $\mathsf{Pr}(y) = 0 \vee \mathsf{Pr}(y) = 1$*, then the above formulas are no longer satisfiable.*

In these examples, we studied the existence of strategies *by determining the satisfiability of formulas. Additionally, we could also determine if some strategies are always better than others, by determining if a certain formula is* valid*.*

Obviously, a GenPSAT instance is also a GGenPSAT instance.

Notice that GGenPSAT has been studied in the context of the decision problem for the probabilistic logic introduced by Fagin, Halpern and Megiddo in [FHM90]. Hence, the computational complexity of this problem is known and addressed in the following theorem.

Theorem 3.1 ([FHM90])**.** GGenPSAT *is* NP*-complete.*

3.4 Reducing GGenPSAT to Satisfiability Modulo Theories

Our goal now is to effectively build a decision procedure for this problem. Previously, in [CCM17b] the authors constructed an effective procedure for the GenPSAT problem by a polynomial reduction to Mixed-Integer Programming (MIP). However, in that framework, one cannot handle Boolean combinations of linear inequalities, at least intuitively. A framework where this problem is naturally expressed is in Satisfiability Modulo Theories (SMT) with respect to the theory of Quantifier-Free Linear Integer and Real Arithmetic (QF_LIRA) [Kin14, BFT16].

Thus, we will now explore the NP-completeness of GGenPSAT and provide a polynomial reduction to QF_LIRA. QF_LIRA is the quantifier-free fragment of the first-order theory that models linear integer and real arithmetic, [Kin14, BFT16]. The variables of the theory can be of one of three sorts: Boolean, integer or real, and the signature is composed of the function symbols $\mathcal{F} = \{0, 1, +, \cdot\}$ and the usual predicates $\{\geq, \leq, <, >, =, \neq\}$. The atoms of the theory are either Boolean variables or linear inequalities involving real and integer variables.

We explore some preliminary steps that lead to an algorithm to solve the GGenPSAT problem.

On the details of the GGenPSAT instance: Assume we are given a GGenPSAT instance (Γ, Ψ), where Γ is a set of classical propositional formulas (hard constraints) $\Gamma = \{\varphi_1, \ldots, \varphi_k\}$ and Ψ is a set of probabilistic formulas (soft constraints) $\Psi = \{\delta_1, \ldots, \delta_s\}$. Recall that a formula δ_j is a Boolean combination of probabilistic atoms of the form

$$q_1 \cdot \mathsf{Pr}(\psi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\psi_\ell) \bowtie q \ ,$$

where $\bowtie \in \{\geq, \leq, <, >, =, \neq\}$.

When ghosts attack: Driven by GenPSAT and PSAT developments, it is simpler to deal with probabilities of propositional variables than with probabilities of propositional formulas. To this end, we introduce *propositional ghost variables* which will represent the propositional formulas occurring inside the probabilistic formulas. Let us define the set of fresh variables that will be used. For this purpose, collect in $\mathsf{InsidePr}(\delta) \subseteq \mathrm{L}_{\mathsf{CPL}}$ all the propositional formulas occurring inside the probabilistic formula $\delta \in \mathsf{Prob}$, which is defined inductively on the structure of δ:

- $\mathsf{InsidePr}(q_1 \cdot \mathsf{Pr}(\psi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\psi_\ell) \bowtie q) = \{\psi_1, \ldots, \psi_\ell\}$;
- $\mathsf{InsidePr}(\neg\delta) = \mathsf{InsidePr}(\delta)$;
- $\mathsf{InsidePr}(\delta_1 \wedge \delta_2) = \mathsf{InsidePr}(\delta_1) \cup \mathsf{InsidePr}(\delta_2)$.

This notion is extended for a set Δ of probabilistic formulas as usual, $\mathsf{InsidePr}(\Delta) = \bigcup_{\delta\in\Delta} \mathsf{InsidePr}(\delta)$. According to this, and recalling that the propositional formulas in Γ need to be satisfied with probability 1, we consider the set of relevant propositional formulas, RelF defined by:

$$\mathsf{RelF} = \Gamma \cup \mathsf{InsidePr}(\Psi) \ .$$

Consider the set of propositional ghost variables corresponding to each element of RelF:

$$\mathfrak{G} = \{\mathfrak{p}_\psi \mid \psi \in \mathsf{RelF}\} \ .$$

Furthermore, we will use the real $[0, 1]$-variable α_ψ to represent the probability of $\psi \in \mathsf{RelF}$.

For ease of notation, we denote by $\mathfrak{G}_i$ the i-th element of $\mathfrak{G}$ and ψ_i the corresponding propositional formula in RelF. We also denote by $|\mathfrak{G}|$ the cardinality of a set $\mathfrak{G}$.

Gathering the propositional variables: The set of propositional variables of interest is $\mathcal{B} = \mathcal{P} \cup \mathfrak{G}$.

Algebraic formulation: Motivated by the algebraic formulation of PSAT and GenPSAT, we express the probabilistic assertions about the elements in RelF algebraically as follows:

$$\begin{cases} V\pi = \alpha \\ \sum \pi_j = 1 \\ \pi \geq 0 \end{cases} \tag{3.3}$$

where:

- $V = [V_{ij}]$ is a matrix of size $|\mathfrak{G}| \times 2^n$, where V_{ij} is defined from the j^{th} valuation $v_j \in \mathcal{V}^*$ and from the i^{th} propositional ghost variable $\mathfrak{G}_i$, by $V_{ij} = v_j(\psi_i)$;
- $\pi = [\pi_j]$ is a vector of size 2^n, where each π_j is a real $[0, 1]$-variable representing the probability valuation v_j;
- $\alpha = [\alpha_i]$ is a vector of size $|\mathfrak{G}|$ and each α_i is a real $[0, 1]$-variable that represents the probability of ψ_i.

Using the following Lemma by Chvátal [Chv83], we can take a step forward in the choice of the right valuations.

Lemma 3.1 ([FHM90, Chv83])**.** *If a system of ℓ linear inequalities with integer coefficients has a nonnegative solution, then it has a nonnegative solution with at most ℓ positive entries.*

Lemma 3.1 tells us that a system with $|\mathfrak{G}|+1$ linear inequalities has a solution iff if has a solution with $|\mathfrak{G}|+1$ nonnegative entries. Furthermore, if a GGenPSAT instance is satisfiable then system (3.3) has a solution. Let us collect in $H = [h_{ij}]$, the $|\mathfrak{G}|+1$ columns of V given by Lemma 3.1, where $h_{|\mathfrak{G}|+1,j} = 1$ for each j, as well as extend α with $\alpha_{|\mathfrak{G}|+1} = 1$, and consider the corresponding probability assignments in π:

$$\begin{cases} H\pi = \alpha \\ \pi \geq 0 \end{cases} . \tag{3.4}$$

When variables multiply: Inspired by the previous arguments, we consider $|\mathfrak{G}|+1$ copies of each propositional variable of interest in $\mathcal{B}$. Each copy is intended to represent the valuations underlying the columns of matrix H. We represent them by

$$\mathcal{B}^{(k)} = \{x^{(k)} \mid x \in \mathcal{P}\} \cup \{\mathfrak{p}^{(k)} \mid \mathfrak{p} \in \mathfrak{G}\} \ .$$

We extend this notation to propositional formulas as expected – given a propositional formula ψ, $\psi^{(k)}$ represents the formula ψ where each of its variables x was replaced by its appropriate copy, $x^{(k)}$. Denote by $\widetilde{\mathcal{B}} = \bigcup_{k=1}^{|\mathfrak{G}|+1} \mathcal{B}^{(k)}$ the set of all copies of all propositional variables.

Probabilistic formulas seen as linear restrictions: To handle probabilistic formulas in the QF_LIRA formalism, we make use of linear inequalities. Since the variable α_ψ represents the probability of each $\psi \in \mathsf{InsidePr}(\Psi)$, we can represent a probabilistic atom $q_1 \cdot \mathsf{Pr}(\psi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\psi_\ell) \bowtie q$ as a linear arithmetic formula of the form $q_1 \cdot \alpha_{\psi_1} + \ldots + q_\ell \cdot \alpha_{\psi_\ell} \bowtie q$. This translation, denoted by PrToLIRA, can be inductively extended to probabilistic formulas (which are Boolean combinations of probabilistic atoms):

- $\mathsf{PrToLIRA}(q_1 \cdot \mathsf{Pr}(\psi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\psi_\ell) \bowtie q)$ is the assertion $q_1 \cdot \alpha_{\psi_1} + \ldots + q_\ell \cdot \alpha_{\psi_\ell} \bowtie q$;
- $\mathsf{PrToLIRA}(\neg\delta)$ is the assertion $\neg\mathsf{PrToLIRA}(\delta)$;
- $\mathsf{PrToLIRA}(\delta_1 \wedge \delta_2)$ is the assertion $\mathsf{PrToLIRA}(\delta_1) \wedge \mathsf{PrToLIRA}(\delta_2)$.

All together now: To verify the satisfiability of the GGenPSAT instance, we will need to satisfy the following constraints:

(hard_constr) $\bigwedge_{\varphi \in \Gamma} \alpha_\varphi = 1$;

(soft_constr) $\bigwedge_{\delta \in \Psi} \mathsf{PrToLIRA}(\delta)$;

(cons) $h_{ik} = 1 \leftrightarrow \mathfrak{G}_i^{(k)}$ for each $i \in \{1, \ldots, |\mathfrak{G}|\}$, $k \in \{1, \ldots, |\mathfrak{G}| + 1\}$;

(val1) $\sum_{j=1}^{|\mathfrak{G}|+1} b_{ij} = \alpha_{\psi_i}$ for each $i \in \{1, \ldots, |\mathfrak{G}|\}$;

(val2) $(0 \leq b_{ij} \leq h_{ij}) \wedge (h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j)$ for each $i \in \{1, \ldots, |\mathfrak{G}|\}$, $j \in \{1, \ldots, |\mathfrak{G}| + 1\}$;

(sums1) $\sum_{j=1}^{|\mathfrak{G}|+1} \pi_j = 1$;

(prop_prob) $\bigwedge_{k=1}^{|\mathfrak{G}|+1} \left(\mathfrak{G}_i^{(k)} \leftrightarrow \psi_i^{(k)}\right)$ for each $i \in \{1, \ldots, |\mathfrak{G}|\}$.

All these restrictions amount to:

- 3 assertions from (hard_constr), (soft_constr) and (sums1);
- $2 \cdot |\mathfrak{G}| \cdot (|\mathfrak{G}| + 1)$ assertions from (cons) and (val2);
- $2 \cdot |\mathfrak{G}|$ assertions from (val1) and (prop_prob).

Hence, we have a total of $\mathcal{O}(|\mathfrak{G}| \cdot (|\mathfrak{G}| + 1))$ assertions, each of polynomial size on the length of (Γ, Ψ) over:

- $|\mathfrak{G}| \cdot (|\mathfrak{G}| + 1)$ binary variables h_{ij};
- $|\mathfrak{G}| \cdot (|\mathfrak{G}| + 1)$ real variables b_{ij};
- $|\mathfrak{G}|$ real variables $0 \leq \alpha_{\psi_i} \leq 1$;
- $|\mathfrak{G}| + 1$ real variables $0 \leq \pi_j \leq 1$;
- $(|\mathfrak{G}| + 1) \cdot (|\mathfrak{G}| + n)$ propositional variables in $\widetilde{\mathcal{B}}$.

With this, we can conclude that the presented procedure translates a GGenPSAT instance into a problem in QF_LIRA of polynomial size.

The solver: We test the satisfiability of a GGenPSAT instance (Γ, Ψ) by translating it to a QF_LIRA problem and then solving the latter appropriately. The procedure presented in Algorithm 4, begins by initializing an empty QF_LIRA problem and uses the following auxiliary procedures:

- assert() introduces an assertion to the QF_LIRA problem;
- PrToLIRA() translates probabilistic formulas into QF_LIRA assertions;
- 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 (Γ, Ψ) is a satisfiable GGenPSAT instance.

Algorithm 4 GGenPSAT solver based on SMT-QF_LIRA

procedure GGenPSAT(props $\{x_i\}_{i=1}^n$, form Γ, probform Ψ)
  assume: $\mathfrak{G} = \{\mathfrak{p}_\psi \mid \psi \in \mathsf{RelF}\}$
  declare: propositional variables: $\widetilde{\mathcal{B}} = \bigcup_{k=1}^{|\mathfrak{G}|+1} \mathcal{B}^{(k)}$
    binary variables: h_{ij} for $i \in \{1, \ldots, |\mathfrak{G}|\}$, $j \in \{1, \ldots, |\mathfrak{G}|+1\}$
    $[0,1]$-variables: α_{ψ_i}, π_j, b_{ij} for $i \in \{1, \ldots, |\mathfrak{G}|\}$, $j \in \{1, ..., |\mathfrak{G}|+1\}$
  for $i = 1$ to $|\mathfrak{G}|$ **do**
    assert($\sum_j b_{ij} = \alpha_{\psi_i}$) ▷ (val1)
    assert($\bigwedge_k (\mathfrak{G}_i^{(k)} \leftrightarrow \psi_i^{(k)})$) ▷ (prop_prob)
    for $j = 1$ to $|\mathfrak{G}|+1$ **do**
      assert($h_{ij} = 1 \leftrightarrow \mathfrak{G}_i^{(j)}$) ▷ (cons)
      assert($0 \leq b_{ij} \leq h_{ij}$) ▷ (val2)
      assert($h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j$) ▷ (val2)
  assert($\bigwedge_\varphi \ \alpha_\varphi = 1$) ▷ (hard_constr)
  assert($\bigwedge_\delta$ PrToLIRA(δ)) ▷ (soft_constr)
  assert($\sum \pi_i = 1$) ▷ (sums1)
  return qf_lira_solver()

Proposition 3.1. *A* GGenPSAT *instance* (Γ, Ψ) *is satisfiable iff Algorithm 4 returns* Sat.

Proof. Assume that a GGenPSAT instance (Γ, Ψ) is satisfiable. Then, there exists a probability distribution ρ over the set of valuations $\mathcal{V}^*$ satisfying (Γ, Ψ). Our goal is to present a model for the QF_LIRA problem obtained by the translation of (Γ, Ψ) describe above. We denote the obtained solutions by $\alpha_\psi^*, \pi_j^*, b_{ij}^*$ and h_{ij}^* and construct a valuation $\widetilde{v}$ over the extended set of propositional variables $\widetilde{\mathcal{B}}$.

For each $\mathfrak{p}_\psi \in \mathfrak{G}$, let α_ψ^* be the probability of the propositional variable $\mathfrak{p}_\psi$ induced by the probability distribution ρ in the following manner:

$$\alpha_\psi^* = \sum_{v : v(\psi)=1} \rho(v) \ . \tag{3.5}$$

Then, consider the algebraic formulation as in (3.3):

$$\begin{cases} V\pi = \alpha^* \\ \sum \pi_j = 1 \\ \pi \geq 0 \end{cases} \tag{3.6}$$

where now the vector $\alpha^* = [\alpha^*_{\psi_i}]$ is defined as in (3.5) and

- $V = [V_{ij}]$ is a matrix of size $|\mathfrak{G}| \times 2^n$, where V_{ij} is defined from the j^{th} valuation $v_j \in \mathcal{V}^*$ and from the i^{th} propositional ghost variable $\mathfrak{G}_i$, by $V_{ij} = v_j(\psi_i)$;
- $\pi = [\pi_j]$ is a vector of size 2^n, where each π_j is a real $[0, 1]$-variable representing the probability valuation v_j.

Note that $\rho^* = [\rho_j]$ where $\rho_j = \rho(v_j)$ is a solution for (3.6). By Lemma 3.1, there exists a matrix $H = [h^*_{ij}]$ composed by $|\mathfrak{G}| + 1$ columns of V such that

$$\begin{cases} H\pi^* = \alpha^* \\ \pi^* \geq 0 \end{cases} \tag{3.7}$$

where $h^*_{|\mathfrak{G}|+1,j} = 1$ for each j and π^* corresponds to the appropriate entries of ρ^*.

Since π^* is also a probability distribution, the assertion (sums1) is satisfied and, considering

$$b^*_{ij} = h^*_{ij} \cdot \pi^*_j \ ,$$

the assertions (val1) and (val2) are also satisfied.

The propositional valuation $\widetilde{v}$ of the variables in $\widetilde{\mathcal{B}}$ inherent to the QF_LIRA model is defined in the following manner:

- $\widetilde{v}(x_i^{(k)}) = v_k(x_i)$;
- $\widetilde{v}(\mathfrak{p}_i^{(k)}) = v_k(\psi_i)$.

This implies that (prop_prob) is satisfied since the valuation $\widetilde{v}$ assigns the same truth value to $\mathfrak{p}_i^{(k)}$ and $\psi_i^{(k)}$. Furthermore, the assertion (cons) is also satisfied as the truth value of $\mathfrak{G}_i^{(k)}$ is given by h^*_{ik}.

Provided that the original probability distribution ρ satisfies the GGenPSAT instance, we immediately conclude that (hard_constr) and (soft_constr) are satisfied, which concludes the proof of the direct implication.

Reciprocally, assume that the associated QF_LIRA problem is satisfiable, and consider the components of its model: a valuation $\widetilde{v}$ of the variables in $\widetilde{\mathcal{B}}$, and

define by y^* the value that the model gives to the variable y. Our aim, is to define a probability distribution ρ over the set of valuations $\mathcal{V}^*$ of the variables in $\mathcal{P}$.

With this purpose, we will refine the valuation $\widetilde{v}$, reduce it to valuations over $\mathcal{B}$, and finally define the probability distribution ρ.

Since $\widetilde{v}$ is a valuation over $\widetilde{\mathcal{B}} = \bigcup_{k=1}^{|\mathfrak{G}|+1} \mathcal{B}^{(k)}$, define its reduct to each copy of $\mathcal{B}$,

$$v_k(p) = \widetilde{v}(p^{(k)}) \text{ , for each } p \in \mathcal{B} \ .$$

Let $W = \{v_1, \ldots, v_{|\mathfrak{G}|+1}\}$ be the set of such valuations. Then, consider the probability distribution $\pi : W \to [0,1]$ defined as $\pi(v_k) = \pi_k^*$. The probability distribution $\rho : \mathcal{V}^* \to [0,1]$ we seek is now easily defined recalling that $\mathcal{B} = \mathcal{P} \cup \mathfrak{G}$:

$$\begin{cases} \rho(v_{i|_{\mathcal{P}}}) = \pi_i^* & \text{for each } i \in \{1, \ldots, |\mathfrak{G}|+1\} \\ \rho(v) = 0 & \text{otherwise} \end{cases}$$

We now need to check that this probability distribution ρ is well defined, i.e., that if $v_{i|_{\mathcal{P}}} = v_{j|_{\mathcal{P}}}$ then $\pi_i^* = \pi_j^*$. We will do this by showing that if $v_i \neq v_j$ then $v_{i|_{\mathcal{P}}} \neq v_{j|_{\mathcal{P}}}$:

- if for some $x \in \mathcal{P}$, $v_i(x) \neq v_j(x)$ then obviously their reducts to $\mathcal{P}$ will also differ in x.

- if for every $x \in \mathcal{P}$, $v_i(x) = v_j(x)$, and there is a $\mathfrak{p} \in \mathfrak{G}$ such that $v_i(\mathfrak{p}) \neq v_j(\mathfrak{p})$ we obtain a contradiction: let ψ be the propositional formula corresponding to $\mathfrak{p}$. Then, since $v_i(x) = v_j(x)$ for every $x \in \mathcal{P}$, then $v_i(\psi) = v_j(\psi)$ which means that $\widetilde{v}(\psi^{(i)}) = \widetilde{v}(\psi^{(j)})$. Since $\widetilde{v}$ satisfies (prop_prob), this means that $\widetilde{v}(\mathfrak{p}^{(i)}) = \widetilde{v}(\mathfrak{p}^{(j)})$ and so $v_i(\mathfrak{p}) = v_j(\mathfrak{p})$.

Since (sums1) is satisfied, ρ constitutes a well-defined probability distribution. To conclude that the hard constraints are satisfied, observe that for each $\varphi \in \Gamma$, since $\alpha_\varphi^* = 1$ it follows that $h_{ij}^* = 1$ for each j such that $\pi_j^* > 0$ and by (cons) and (prop_prob) it means that ρ satisfies φ with probability 1. For soft constraints, the reasoning is similar – observe that (cons) and (prop_prob) links the algebraic reasoning with the valuations. To show that ρ satisfies Ψ, i.e., the soft constraints, since the assertion (soft_constr) is satisfied, it is enough to show that $\mathsf{Pr}(\psi)$ coincides with α_i^*, where ψ is the i-th formula of $\mathsf{InsidePr}(\Psi)$. In fact, the probability of ψ is

$$
\begin{aligned}
\mathsf{Pr}(\psi) &= \sum_{v \in \mathcal{V}^*} v(\psi) \cdot \rho(v) \\
&= \sum_{v \in W} v(\psi) \cdot \rho(v) \\
&= \sum_{j=1}^{|\mathfrak{G}|+1} v_j(\psi) \cdot \pi_j^* \\
&= \sum_{j=1}^{|\mathfrak{G}|+1} \widetilde{v}(\psi^{(j)}) \cdot \pi_j^* \\
&= \sum_{j=1}^{|\mathfrak{G}|+1} h_{ij}^* \cdot \pi_j^* \\
&= \sum_{j=1}^{|\mathfrak{G}|+1} b_{ij}^* && \triangleright \text{ by } (\mathsf{val2}) \text{ assertions} \\
&= \alpha_i^* \ .
\end{aligned}
$$

We detail the second to last step of the deduction: either h_{ij}^* is 0 and by the first (val2) assertion we obtain that b_{ij}^* is also 0 and so $h_{ij}^* \cdot \pi_j^* = 0 = b_{ij}^*$; or h_{ij}^* is 1 and by the (val2) assertions we obtain that b_{ij}^* equal to π_j^* and so $h_{ij}^* \cdot \pi_j^* = 1 \cdot \pi_j^* = b_{ij}^*$.

This shows that ρ satisfies the formulas in Ψ and so it is a model for the GGenPSAT instance (Γ, Ψ). □

3.5 Phase Transition

In this section we describe the tool we developed to implement the algorithm that solves the GGenPSAT problem. With this in hands, we generate batches of random GGenPSAT instances and study the behaviour of the implemented solver, in terms of time and satisfiability.

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 Python, and we used Yices [Dut14], version 2.5.1, to solve the SMT problem. Our tool takes as input a GGenPSAT written in an (smt-lib)-style notation enriched with the probability operator `(pr` φ`)`. As an example, the problem from Example 3.2 can be formulated as

```
(define x::bool)
(define y::bool)
```

```
(assert (>= (pr (not (xor x y))) (* 2 (pr (xor x y)))))
(assert (> (pr (xor x y)) 0))
(assert (or (= (pr y) 0) (= (pr y) 1)))
(assert (= (pr x) 0))
(check)
```

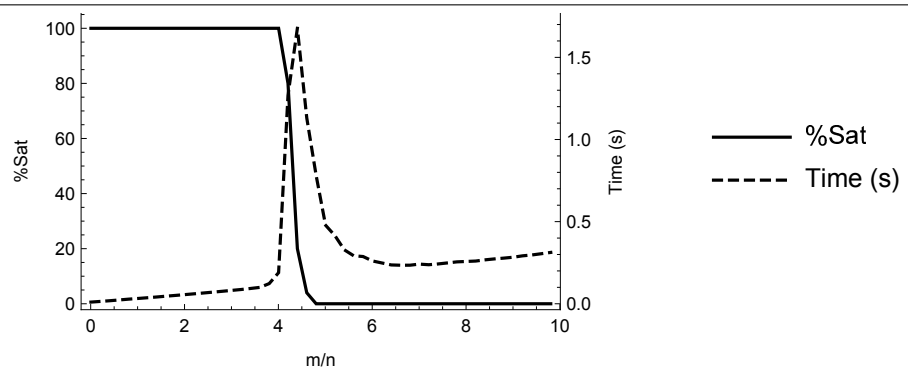
The machine used for the tests was a Mac Pro at 3.33 GHz 6-Core Intel Xeon with 6 GB of memory. Our implementation is available in [CMC16b].

As we have described in the last chapter, a phase transition behaviour is characterized by a sharp transition between two clearly distinct states. Regarding satisfiability problems, these states correspond to states in which the problems are either satisfiable or not satisfiable. In [GW94], this behaviour was studied for random 3SAT problems and, heuristically, shown that the ratio m/n of number of clauses over the number of variables characterizes the phase transition. That is, there is a value (close to 4.3 on 3SAT) of m/n for which the random problems rapidly transition from being satisfiable to not being satisfiable. Furthermore, it is usually during this phase transition that the harder random instances lie, and this can heuristically be observed by a peak in time taken to solve problems in this critical area.

We begin by studying the behaviour of the GGenPSAT when dealing with random 3SAT, PSAT and GenPSAT instances. Besides studying the phase transition behaviour, this is also aimed at determining whether the GGenPSAT translation has a big overhead in terms of solving efficiency. In fact, our results show that the presented solver is more efficient than the previously developed solver for GenPSAT, [CMC16a]. This is not unexpected since the translation used here for GGenPSAT is much more natural and concise than the one used for GenPSAT.

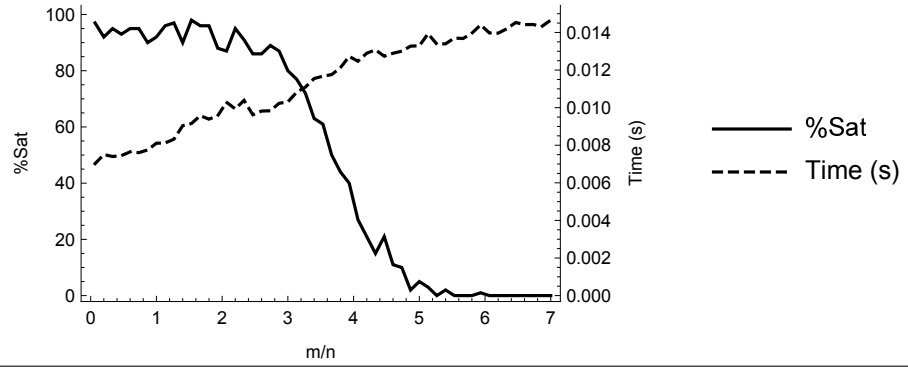
We denote by n the number of variables, m the number of propositional clauses and k the number of probabilistic clauses of a problem and generate 100 random formulas for each data point. In random 3SAT instances, we observe the phase transition behaviour when m/n is close to 4.3 as previously detected in [GW94], see Figure 3.2.

Figure 3.2 Phase transition for 3SAT seen as a GGenPSAT instance, with $n = 200$.



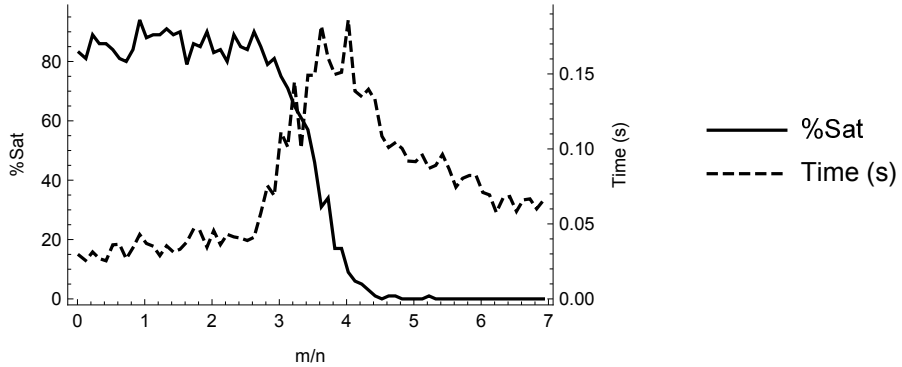
Regarding PSAT, we encode random PSAT instances in GGenPSAT with the same parameters as done in [CCM17b], $n = 15$ and $k = 2$. In Figure 3.3, we can clearly observe a phase transition, however no performance decline is observed during it. This can be explained by the efficient translation from GGenPSAT to SMT as well as the performance of Yices. Note that the running times are negligible.

Figure 3.3 PSAT phase transition seen as a GGenPSAT instance, with $n = 15$ and $k = 2$.



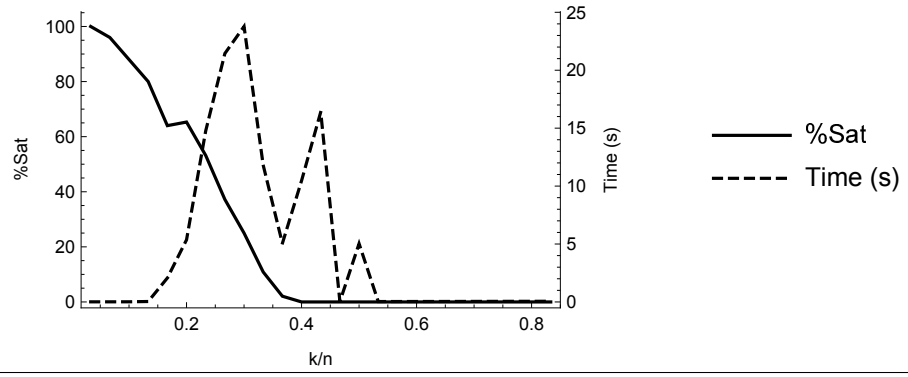
By increasing the number of variables and probabilistic formulas to $n = 40$ and $k = 4$, we can now detect the decrease in performance in Figure 3.4.

Figure 3.4 PSAT phase transition seen as a GGenPSAT instance, with $n = 40$ and $k = 4$.



We also study the phase transition of random PSAT instances by studying the ratio k/n in Figure 3.5. As in [CCM17b], we obtain very similar results but, again, the performance of this translation is much better – with a peak of $25s$ during the phase transition *vs* $60s$ in [CCM17b].

Figure 3.5 PSAT phase transition as a GGenPSAT instance, with $n = 30$ and $m = 40$.



We also obtain very similar results for the embedding of GenPSAT in GGenPSAT, seen in Figures 3.6, 3.7 – as expected the phase transitions stay the same as previously detected, and a performance boost is gained by the efficient GGenPSAT to SMT translation.

Figure 3.6 GenPSAT phase transition as a GGenPSAT instance, with $n = 15$ and $k = 2$.

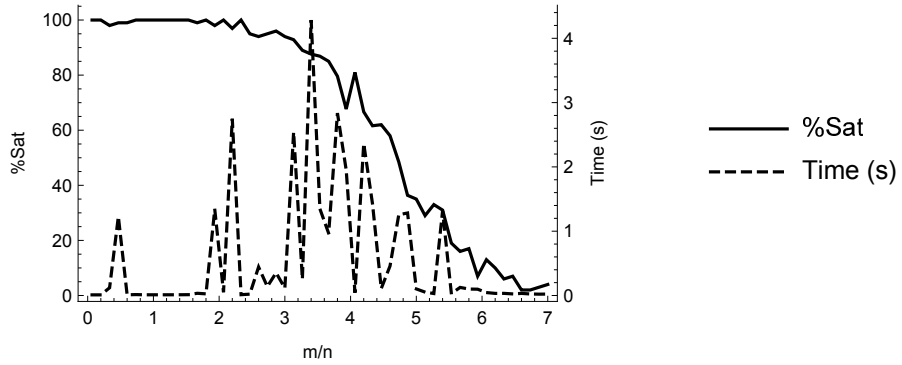
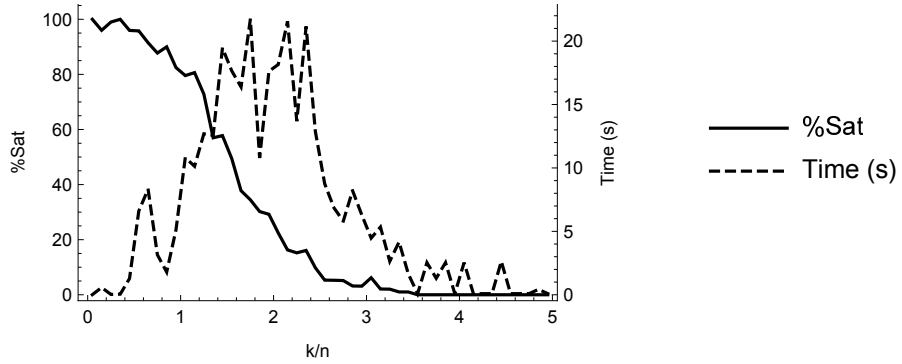


Figure 3.7 GenPSAT phase transition as a GGenPSAT instance, with $n = 20$ and $m = 10$.



As expected, being an NP-complete problem, the full GGenPSAT problem also exhibits a phase transition behaviour. We generated the random GGenPSAT instances as follows: a random 3SAT instance with n variables and m clauses is generated and then, each variable x_i is replaced by a problem G_i which is a conjunction of m random probabilistic atoms over n variables. The results of those tests are seen in Figures 3.8, 3.9.

Figure 3.8 Random GGenPSAT instances with $n = 30$ and $k = 2$.

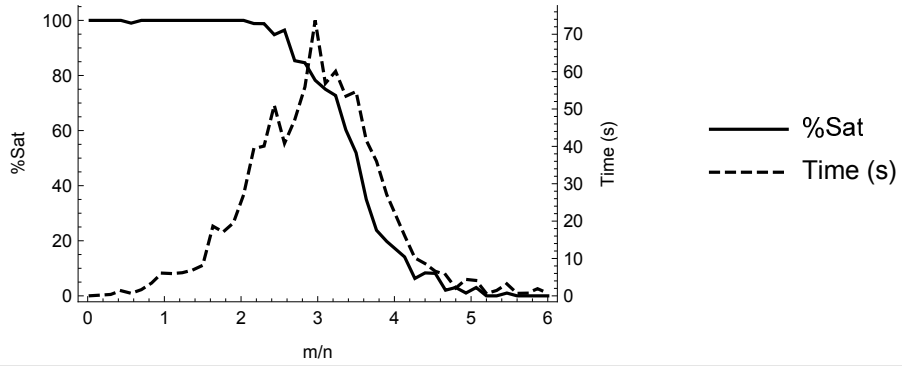
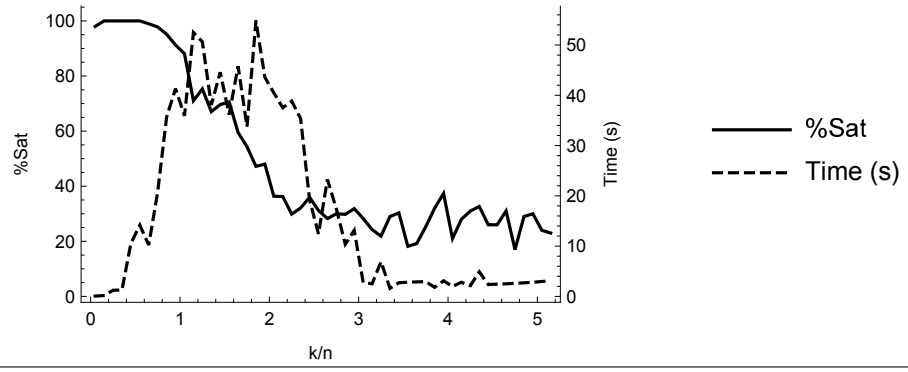


Figure 3.9 Random GGenPSAT instances with $n = 20$ and $m = 10$.



In summary, we believe the developed tool (due to the size of the translation) is able to solve reasonably sized instances, surpassing as well the GenPSAT dedicated tool developed in [CCM17b] for those instances. Given this, we are able to detect the phase transition behaviour and heuristically determine parameter ratios for which random instances are hard to solve.

3.6 Applications

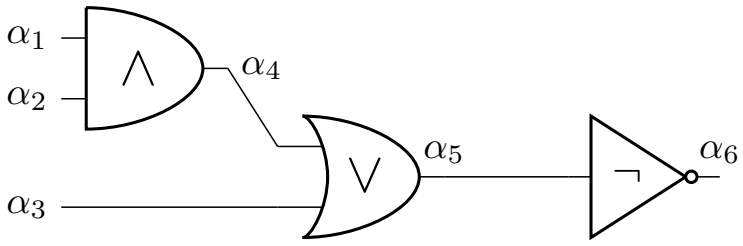
In this section we will showcase some applications of the GGenPSAT formalism. In particular we will model examples from hardware verification and side-channel attacks in the GGenPSAT mindset and actually provide specifications of instances

of problems that can be run on the GGenPSAT tool, [CMC16b]. Other examples not described here can also be found in [CMC16b] such as the zero-knowledge example provided in [MSS05].

3.6.1 Hardware verification

Formal verification in general has greatly impacted hardware verification, allowing the minimization of circuit sizes, bug finding and solving general design problems. With a probabilistic formalism, we are also able to model unreliable circuits, as well as determine if a certain circuit satisfies a safety guarantee, e.g., work 96% of the time as expected.

The first example is from [Bal10] and studies the implementation of a circuit for an 2-1 AND-OR-INVERTER (AOI21). An AOI21 is the function on 3 bits given by $\mathsf{AOI21}(\alpha_1, \alpha_2, \alpha_3) = \neg((\alpha_1 \wedge \alpha_2) \vee \alpha_3)$. A circuit for this function is for instance the following:



whose implementation is given by the formula

$$\mathsf{Impl} \triangleq (\alpha_4 \leftrightarrow \alpha_1 \wedge \alpha_2) \wedge (\alpha_5 \leftrightarrow \alpha_3 \vee \alpha_4) \wedge (\alpha_6 \leftrightarrow \neg\alpha_5) \ .$$

The validity of this implementation can be easily checked by verifying that

$$\mathsf{Impl} \wedge \neg(\alpha_6 \leftrightarrow \mathsf{AOI21}(\alpha_1, \alpha_2, \alpha_3))$$

is unsatisfiable. In GGenPSAT this can be done in the following way

```
(define a1::bool)
(define a2::bool)
(define a3::bool)
(define a4::bool)
(define a5::bool)
(define a6::bool)

(assertprop (and
```

```
                (and (<=> a4 (and a1 a2))
                     (<=> a5 (or a3 a4))
                     (<=> a6 (not a5)))
                (not (<=> a6 (not (or (and a1 a2) a3))))))
(check)
```

More interestingly, we might want to study the reliability of the whole circuit, depending on the reliability of each individual gate. For instance, suppose that the $\wedge$-gate works as expected at least 97% of the time, the $\vee$-gate works at least 99% of the time and the $\neg$-gate always produces the expected value. This description of the circuit can be formalized in the probabilistic logic language as

$$\widetilde{\mathsf{Impl}} \triangleq \mathsf{Pr}(\alpha_4 \leftrightarrow \alpha_1 \wedge \alpha_2) \geq 0.97 \wedge \mathsf{Pr}(\alpha_5 \leftrightarrow \alpha_3 \vee \alpha_4) \geq 0.99 \wedge \mathsf{Pr}(\alpha_6 \leftrightarrow \neg\alpha_5) = 1 \ .$$

Suppose we now would like to guarantee that this implementation in fact computes the AOI21 function at least 96% of the time, under the previous assumptions of the faulty gates. This is modelled by the following formula

$$\widetilde{\mathsf{Spec}} \triangleq \mathsf{Pr}(\alpha_6 \leftrightarrow \neg((\alpha_1 \wedge \alpha_2) \vee \alpha_3)) \geq 0.96 \ .$$

Hence, to guarantee that under the reliability of the gates we reach this performance, we need to check the satisfiability of

$$\widetilde{\mathsf{Impl}} \wedge \neg\widetilde{\mathsf{Spec}} \ .$$

If it is not satisfiable, we are guaranteed that indeed this circuit works as an AOI21 at least 96% of the time. This can be formally verified in GGenPSAT as

```
(define a1::bool)
(define a2::bool)
(define a3::bool)
(define a4::bool)
(define a5::bool)
(define a6::bool)

(assert (> (pr (<=> a4 (and a1 a2))) 0.97))
(assert (> (pr (<=> a5 (or a3 a4))) 0.99))
(assertprop (<=> a6 (not a5)))

(assert (not (>= (pr (<=> a6 (not (or a3 (and a1 a2))))) 0.96)))
(check)
```

which indeed is an unsatisfiable set of formulas.

Furthermore, notice how this example does not make full use of the expressiveness of the whole probabilistic logic. In particular, we do not make use of linear combinations of probabilistic terms. To showcase this, we present another example which will motivate the following chapter on applications to side-channel analysis.

3.6.2 Boolean masking

Consider a circuit with 3 Boolean inputs which computes the function $\varphi(k, r_1, r_2) = k \oplus (r_1 \oplus r_2)$, where k is a secret that is to be masked using the exclusive or, $\oplus$, of two independent $\mathsf{Bernoulli}(\frac{1}{2})$ random Boolean variables r_1, r_2. Our goal is to determine if this mask actually works, i.e., whether it reveals the value of k if we sample the value of $\varphi(k, r_1, r_2)$ enough times. For this, we consider the probability of the circuit returning 1 depending on the value of k. If this probability differs with k, we can sample the circuit to determine k.

To model this problem, we need to find two keys k, k' such that the probability of the formula $\varphi(k, r_1, r_2)$ differs from the probability of $\varphi(k', r_1, r_2)$ and thus forcing that $\mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2))$. To define each key as fixed but unknown, we enforce that $\mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1$ and $\mathsf{Pr}(k') = 0 \vee \mathsf{Pr}(k') = 1$. Modelling $\mathsf{Bernoulli}(\frac{1}{2})$ random variables is simple in GGenPSAT, $\mathsf{Pr}(r_i) = \frac{1}{2}$ for $i = 1, 2$. Regarding the independence of r_1 and r_2 we should impose that $\mathsf{Pr}(r_1 \wedge r_2) = \mathsf{Pr}(r_1)\mathsf{Pr}(r_2)$ which is not possible since the language does not have products of probabilistic terms. However, when the probability of each random variable $\mathsf{Pr}(r_i)$ is known this can always be expressed, despite leading to an exponential number of formulas [Mor17]. Thus, independence can be modelled as $\mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4}$. Finally, the whole problem can be prescribed in GGenPSAT as the following set of assertions:

$$\begin{cases} \mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2)) \\ \mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1 \\ \mathsf{Pr}(k') = 0 \vee \mathsf{Pr}(k') = 1 \\ \mathsf{Pr}(r_i) = \frac{1}{2} \qquad \text{for } i = 1, 2 \\ \mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4} \end{cases}$$

The encoding in the GGenPSAT tool is straightforward,

```
(define k::bool)
(define kp::bool)
(define r1::bool)
(define r2::bool)

(assert (not (= (pr (xor k (xor r1 r2))) (pr (xor kp (xor r1 r2))))))

(assert (or (= (pr k) 0) (= (pr k) 1)))
```

```
(assert (or (= (pr kp) 0) (= (pr kp) 1)))

(assert (= (pr r1) (/ 1 2)))
(assert (= (pr r2) (/ 1 2)))
(assert (= (pr (and r1 r2)) (/ 1 4)))
(check)
```

and we obtain that the set of assertions is unsatisfiable. This means that indeed, this formula is secure under independent $\mathsf{Bernoulli}(\frac{1}{2})$ random masks, since there is no model in which the weight of the formula $k \oplus (r_1 \oplus r_2)$ depends on the value of k. However, if we drop the independence restriction, $\mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4}$, we obtain a satisfiable instance and thus, the circuit leaks information about the secret key, e.g. $r_1 = r_2$ implies $\varphi(k, r_1, r_2) = k$.

We will further explore this topic in the next chapter, Chapter 4, where we consider active attackers with side-channel capabilities.

3.7 Conclusions and Future Work

In this chapter, we aimed to study a generalization of the probabilistic satisfiability problem. The GGenPSAT problem naturally models Boolean combinations of linear inequalities involving probabilities of propositional formulas. We developed a satisfiability procedure by a reduction to the quantifier-free theory of integer and real arithmetic, and proved its correctness. Furthermore, we implemented a tool that translates problems in GGenPSAT to QF_LIRA and solves them with an off-the-shelf SMT solver such as Yices. With this tool in hands we are able to detect and study the phase-transition behaviour of this problem. It is also worth noting that since the expressiveness of the GGenPSAT problem coincides with the probabilistic logic of Fagin et al., [FHM90], this tool also serves as a satisfiability procedure for the logic.

We believe the study of this problem and subsequent tool implementation provided a sound foundational basis to the development of applications where probabilistic reasoning is required. This will become clear in the next chapter, where we model very naturally the problem of perfect masking in side-channel attacks in this probabilistic formalism.

Chapter 4

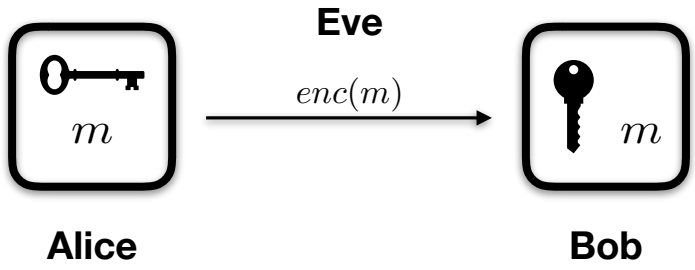
A Probabilistic Formalization of Attackers with Side-Channel Capabilities

4.1 Introduction

Even cryptographic protocols which are based on mathematically hard to solve problems can be easily exploited when enough "physical" information is leaked to the outside by the system implementing it – this was the lesson taught by the seminal paper on the power of side-channel attacks, [Koc96]. Side-channel attacks happen when an attacker is able to obtain supposedly private data through information that the system leaks via physical channels such as timing data [Koc96], power consumption [KJJ99], electromagnetic radiation [GMO01, QS01, AARR03], temperature [HS13] to name some impactful side channels. These channels can also be used to exfiltrate data in a covert manner, where an attacker is able to encode data in these channels. With such huge attack surface, this area has not stopped developing and has showed that commonly used implementations of cryptographical protocols and primitives are not safe against these attacks, RSA, DSS and Diffie-Hellman [Koc96], elliptic curve implementations in the GnuPG's Libgcrypt [GPPT16] as well as symmetric encryption schemes such as DES [KJJ99] and AES [Ram17]. These attacks even extend to quantum protocols such as quantum key distribution [LLK07, NFSM$^+$09], which in theory are physically secure.

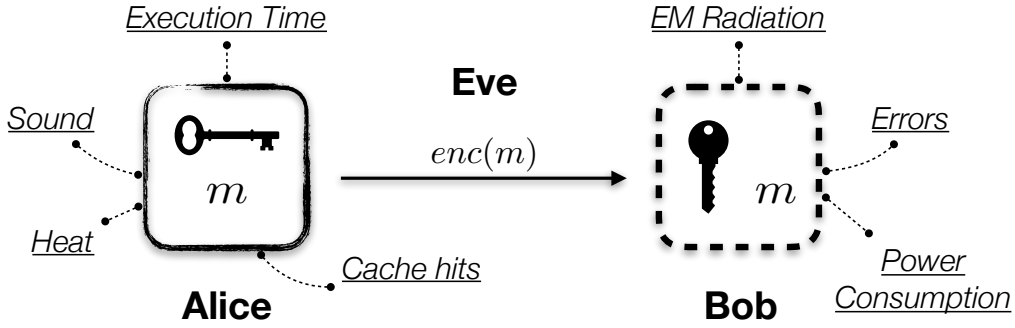
Despite the security guarantees of the protocols, often enough, their security proofs do not usually take into account the information leaks the system may have through physical properties. In the traditional cryptography view of the world, an attacker only observes the public part of the protocol. For instance, in a private-key encryption scenario Eve, the attacker, would only have access to the encryption of the message exchanged between Alice and Bob, see Figure 4.1.

Figure 4.1 The traditional cryptography view of the world.



However, in the real-world, the encryption process takes time, the machine in which the encryption is being made consumes power, emits heat, electromagnetic radiation and sound, as depicted in Figure 4.2. Unless defensive measures are taken into account more than one of these channels of physical information about the system may be available to an attacker. Furthermore, the attacker can actively force information leaks by injecting or forcing faults in the system. These fault attacks, introduced in [BDL97] and extended in [BDL01], can often lead to full secret recovery, even on standard ciphers such as AES [DLV03].

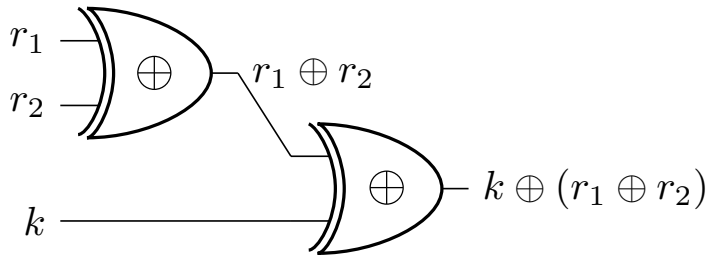
Figure 4.2 The side-channel view of the world.



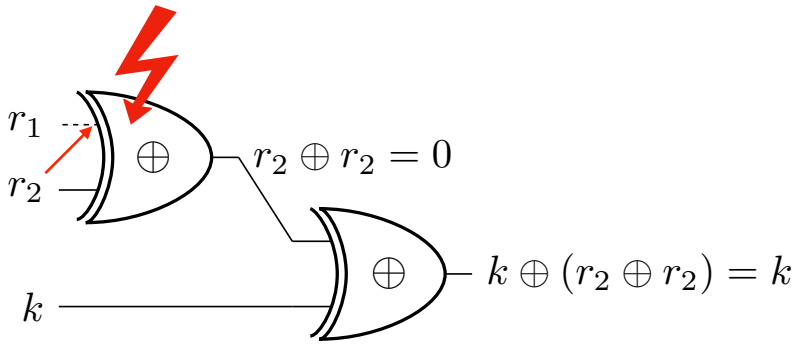
There are some usual approaches to thwart these attacks: on one hand there can be a physical shielding of the device running the cryptographic functions, trying to prevent unwanted leakage of information; on the other hand, there can be a logical shielding of the secrets, e.g., by means of a random mask which is applied to values during the execution of algorithm [CJRR99, PR13]. In this chapter, we study the latter case, that is, how to develop algorithms with a provably secure implementation. We also only focus on power side-channel attacks – for instance timing side-channel attacks have other countermeasures such as constant-time code, which are out of the scope of this work.

In this chapter, we use the probabilistic propositional logic [FHM90], and its associated satisfiability problem, to formalize the notion of perfect masking of a Boolean formula from [EWS14]. Furthermore, we generalize the notion of perfect masking to encompass an active attacker interfering with the system, and consider the problem of deciding whether a formula is perfectly masked under the attack of such an adversary (or a general family of attackers).

For this, we consider two generic families of active attackers: *attackers with fault-injection capabilities* which are able to partially (or fully) control the masks being used to protect the leakage of information and *attackers with variable-dependency capabilities* which are able to make two or more, previously independent random variables, dependent. The latter attacker can seem devoid of practical consequence but consider the following scenario in a simple circuit where a secret bit k is being masked by two independent and $\mathsf{Bernoulli}(\frac{1}{2})$ random variables r_1, r_2 using xor, $k \oplus (r_1 \oplus r_2)$.



Suppose now that the attacker is able to physically interfere with the wiring of the circuits in a way that the r_1 variable is sometimes overridden by r_2:



In this scenario, when the fault occurs, the output of the circuit will be $k \oplus (r_2 \oplus r_2) = k$, i.e., the mask will fail.

Other formal method techniques have been employed to study the effectiveness of masking to prevent side-channel attacks. The authors of [ISW03] propose several

techniques to transform a circuit into another that tolerates high order probing attacks. More in line with our approach is the sensitivity method of [BRNI13]. However, as shown in [EWS14] this metric fails to account for some masking vulnerabilities. As such, we mainly follow the framework of [EWS14] and their perfect masking notion. The same authors also propose a way to synthesize masks in [EW14]. Recent developments studying higher order masking can be found in [BBD+15].

The main contribution of this chapter is the characterization of attackers with side-channel capabilities in a probabilistic formalism. With these tools we formalize the notion of a perfectly masked circuit against such attackers. Surprisingly, when facing a very powerful attacker, this problem actually becomes easier and is shown to be in co-NP.

4.2 Preliminaries

We will begin by recalling preliminary notions of propositional logic and GGenPSAT from the previous chapter. Then, we will introduce basic side-channel notions as well as define the perfect masking decision problem, the main focus of this chapter.

4.2.1 Propositional logic and GGenPSAT

Let us begin by fixing a set of propositional variables $\mathcal{P} = \{x_1, \ldots, x_n\}$. The set of *classical propositional formulas* is defined, as usual, by

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

A *propositional literal* is either a propositional variable or its negation. A *propositional clause* is a non-empty disjunction of one or more propositional literals. We also denote the *size of a classical propositional formula* φ by $|\varphi|$, and is inductively defined as follows:

- $|x| = 1$ for $x \in \mathcal{P}$;
- $|c(\varphi_1, \ldots, \varphi_n)| = 1 + |\varphi_1| + \ldots + |\varphi_n|$, where c is an n-ary connective and $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$.

A *propositional valuation* is a map $v : \mathcal{P} \to \{0, 1\}$, which is extended to propositional formulas as usual. We say that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$. This notion is extended to sets of propositional formulas as usual. Let $\mathcal{V}^* = \{v_1, \ldots, v_{2^n}\}$ be the set of all valuations defined over variables of $\mathcal{P}$. We define a *probability distribution* π over $\mathcal{V}^*$ as a probability vector of size 2^n.

We recall from [FHM90] the set of *probabilistic atoms* (used herein to define probabilistic formulas) composed by linear inequalities of probabilities of propositional formulas with rational coefficients:

$$\mathsf{PAt} ::= \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L}_{\mathsf{CPL}}) + \ldots + \mathbb{Q} \cdot \mathsf{Pr}(\mathrm{L}_{\mathsf{CPL}}) \geq \mathbb{Q} \ .$$

The set of *probabilistic formulas* is defined as a Boolean combination of probabilistic atoms as follows:

$$\mathsf{Prob} ::= \mathsf{PAt} \mid \neg\mathsf{Prob} \mid \mathsf{Prob} \wedge \mathsf{Prob} \ .$$

Observe that the other relational symbols $\{<, >, \leq, =, \neq\}$ can be defined by abbreviation, as well as the logical connectives $\rightarrow, \vee, \leftrightarrow$.

To interpret probabilistic formulas, we consider a probability distribution π over $\mathcal{V}^*$. The satisfaction relation is inductively defined as:

- $\pi \Vdash q_1 \cdot \mathsf{Pr}(\varphi_1) + \ldots + q_\ell \cdot \mathsf{Pr}(\varphi_\ell) \geq q$ iff $\sum_{i=1}^{\ell} \left(q_i \left(\sum_{j=1}^{2^n} v_j(\varphi_i) \cdot \pi_j \right) \right) \geq q$;
- $\pi \Vdash \neg\delta$ iff $\pi \nVdash \delta$;
- $\pi \Vdash \delta_1 \wedge \delta_2$ iff $\pi \Vdash \delta_1$ and $\pi \Vdash \delta_2$,

where $\delta, \delta_1, \delta_2 \in \mathsf{Prob}$, $q, q_i \in \mathbb{Q}$ and $\varphi_i \in \mathrm{L}_{\mathsf{CPL}}$ where $i \in \{1, \ldots, \ell\}$. A probability distribution π *satisfies* $\delta \in \mathsf{Prob}$ if $\pi \Vdash \delta$ and satisfies a set of probabilistic formulas if it satisfies each one of them.

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

$$\Xi_{(\Gamma,\Psi)} = \Psi \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Gamma\} \ . \tag{4.1}$$

Definition 4.1 (GGenPSAT problem)**.** *Given a* GGenPSAT *instance* (Γ, Ψ)*, the Classical Generalized Probabilistic Satisfiability problem* (GGenPSAT) *consists in determining whether there exists a probability distribution* π *over* $\mathcal{V}^*$ *that satisfies* (Γ, Ψ)*.*

4.2.2 Side-channel

Throughout this section, we are focused in studying the following scenario: there is a Boolean formula φ with *plaintext* variables $x \in X$, *secret key* variables $k \in K$,

and *random mask* variables $r \in R$, that computes a *ciphertext* $\varphi(X, K, R)$. Typically, the Boolean masks are independently sampled according to a $\mathsf{Bernoulli}(\frac{1}{2})$ distribution, however in general this might not happen. For this, assume that the masks are sampled according to a probability distribution $\mathbb{P} : \{0,1\}^{|R|} \to [0,1]$, where $|R|$ denotes the cardinality of the set R.

Given a Boolean formula φ with sets of variables X, Y we may denote it by $\varphi(X, Y)$ to reinforce that X, Y are free variables in φ. Furthermore, an instantiation of the variables in X is usually denoted by a bold face letter $\mathbf{X} \in \{0,1\}^{|X|}$. Thus, $\varphi(X, Y)$ where the variables in X are instantiated to $\mathbf{X}$ is denoted by $\varphi(\mathbf{X}, Y)$. We also denote $\{0,1\}^{|X|}$ by $\mathsf{dom}(X)$.

We say that a set S has *polynomial size* in φ, denoted by $|S| = \mathsf{poly}(|\varphi|)$, when there is a positive polynomial p such that $|S| \leq p(|\varphi|)$ and also that $|\psi| \leq p(|\varphi|)$ for all formulas $\psi \in S$.

We denote by $\wp(S)$ the set of subsets of S and by $\wp_{\geq 2}(S)$ the set of subsets of S with cardinality greater than 2.

Definition 4.2 (Induced Probability Distribution)**.** *Given a Boolean formula $\varphi(X, K, R)$, and $\mathbb{P}$ the probability distribution of R, we denote the* probability distribution induced *by φ with $D_{\mathbb{P},\varphi} : \mathsf{dom}(X) \times \mathsf{dom}(K) \to [0,1]$. Specifically, given a valuation on the plaintexts and keys, $v : X \cup K \to \{0,1\}$, the induced distribution $D_{\mathbb{P},\varphi}(v(X), v(K))$ is a random variable D with probability*

$$\mathsf{P}(D = 1) = \sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_{\mathbf{R}}(\varphi) \ ,$$

where $\bar{v}_{\mathbf{R}} : X \cup K \cup R \to \{0,1\}$ extends the valuation v to R as prescribed by $\mathbf{R} \in \mathsf{dom}(R)$.

Example 4.1. *Consider the Boolean formula $\varphi \triangleq x \oplus k \oplus (r_1 \oplus r_2)$ and assume that the random variables are $\mathsf{Bernoulli}(\frac{1}{2})$, and independently generated. Then, the induced probability distribution D for the valuation $v(x) = 0, v(k) = 1$ is*

$$\begin{aligned}
\mathsf{P}(D = 1) &= \sum_{\mathbf{R} \in \{0,1\}^2} \frac{1}{4} \cdot \bar{v}_{\mathbf{R}}(\varphi) \\
&= \frac{1}{4} \cdot \bar{v}_{(1,1)}(\varphi) + \frac{1}{4} \cdot \bar{v}_{(0,0)}(\varphi) + \frac{1}{4} \cdot \bar{v}_{(1,0)}(\varphi) + \frac{1}{4} \cdot \bar{v}_{(0,1)}(\varphi) \\
&= \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 0 \\
&= \frac{1}{2} \ .
\end{aligned}$$

Definition 4.3 (Perfect Masking)**.** *A Boolean formula φ is* perfectly masked *under $\mathbb{P}$ if*

$$D_{\mathbb{P},\varphi}(\mathbf{X}, \mathbf{K}) = D_{\mathbb{P},\varphi}(\mathbf{X}, \mathbf{K}') \ ,$$

for any plaintext $\mathbf{X}$ and secret keys $\mathbf{K}, \mathbf{K}'$.

Consider the following illustrative example of these concepts.

Example 4.2 ([EWS14])**.** *Consider the following masking functions and their respective output given the values of k, r_1 and r_2 assuming that r_1, r_2 are independent and* Bernoulli($\frac{1}{2}$) *randomly sampled as defined in Table 4.1.*

$$\begin{cases} o_1 = k \wedge (r_1 \wedge r_2) \\ o_2 = k \vee (r_1 \wedge r_2) \\ o_3 = k \oplus (r_1 \wedge r_2) \\ o_4 = k \oplus (r_1 \oplus r_2) \end{cases}$$

k	r_1	r_2	o_1	o_2	o_3	o_4
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	1	0	1

Table 4.1: Boolean masking examples: only o_4 perfectly masks the secret value k.

By inspection of the output probability distribution we observe that it varies depending on k for all masking functions except for o_4. This is the only function in which the output distribution is the same for both $k = 1$ and $k = 0$. An attacker sampling the function would observe the same ratio of zeros and ones independently of the value k, which is not true for the other masking functions.

Then the problem we want to solve is naturally defined:

Definition 4.4 (Perfect masking decision problem)**.** *Given a Boolean formula φ with plaintexts X, secret keys K and random variables R distributed according to $\mathbb{P}$, determine if φ is perfectly masked under $\mathbb{P}$.*

We also remark that this approach to study power side-channel masks implicitly assumes a *Hamming Weight* model for the attacker abilities. This means that an attacker is able to successfully distinguish values with different Hamming weights which is a common model used to study power-related side-channel attacks since they correlate well enough [MOP07]: a computation on variables with a high Hamming weight consumes more power than a computation on variables with a low Hamming weight.

4.3 Modelling perfect masking in GGenPSAT

A formula is perfectly masked if it is not possible to distinguish the distributions on the outputs when different keys $\mathbf{K}, \mathbf{K}'$ are being used. This depends on the Boolean formula φ which encodes the computation or circuit, as well as the joint

probability distribution of the random masks $r \in R$. In the case that the joint probability distribution can be modelled in a finite set of GGenPSAT formulas, we denote it by $\mathcal{R}(\mathbb{P})$. Then, in the probabilistic formalism of GGenPSAT, testing if a formula is perfectly masked corresponds to determining if the set of formulas,

$$\begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) \\ \mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1 \text{ for } x \in X \\ \mathsf{Pr}(k) = 0 \vee \mathsf{Pr}(k) = 1 \text{ for } k \in K \cup K' \\ \mathcal{R}(\mathbb{P}) \end{cases} \tag{4.2}$$

is unsatisfiable, where $\mathcal{R}(\mathbb{P})$ is a set of GGenPSAT formulas that model the probability distribution $\mathbb{P}$ of the random masks. Notice that the case where this formula is unsatisfiable is when there do not exist variables $\mathbf{X}, \mathbf{K}, \mathbf{K}'$ for which the probabilities $\mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R))$ and $\mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K}',R))$ do not coincide and so, the probability distribution is indistinguishable, thus making the computation $\varphi(X,K,R)$ secure i.e., φ is perfectly masked in the sense of [EWS14].

In this sense, it is natural to model the problem of deciding if a computation is perfectly masked as a GGenPSAT problem. Each instance is composed by a set of formulas (4.2) mainly parametrized by a set of probabilistic formulas $\mathcal{R}(\mathbb{P})$ specifying the probability distribution on the masks to be considered.

As we will make extensive use of the probabilistic formula scheme of the form $\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1$, which states that x is essentially propositional in nature, we introduce it as an abbreviation

$$\mathsf{prop}(x) \triangleq (\mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1) \ ,$$

which we also extend to sets of variables X in the natural way

$$\mathsf{prop}(X) \triangleq (\bigwedge_{x \in X} \mathsf{Pr}(x) = 0 \vee \mathsf{Pr}(x) = 1) \ .$$

This way, we restate the GGenPSAT formulation (4.2) as

$$\mathsf{PM}(\varphi,\mathbb{P}) : \begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathcal{R}(\mathbb{P}) \end{cases} \tag{4.3}$$

where $\mathcal{R}(\mathbb{P})$ uniquely characterizes the joint probability distribution $\mathbb{P}$ of the random variables in R. We now prove that indeed this formulation corresponds to the problem of deciding if a Boolean formula is perfectly masked under $\mathbb{P}$. For this, we first need an auxiliary lemma.

Lemma 4.1. *Let $\psi(x, Y)$ be a Boolean formula with free variables $\{x\} \cup Y$. The following implications hold:*

1. $\mathsf{Pr}(x) = 0 \rightarrow \mathsf{Pr}(\psi(x, Y)) = \mathsf{Pr}(\psi(0, Y))$;

2. $\mathsf{Pr}(x) = 1 \rightarrow \mathsf{Pr}(\psi(x, Y)) = \mathsf{Pr}(\psi(1, Y))$.

Proof. To prove case *1.* assume that $\mathsf{Pr}(x) = 0$ and observe that, denoting by π the probability distribution over the valuations on $\{x\} \cup Y$,

$$\begin{aligned}
\mathsf{Pr}(\psi(x, Y)) &= \sum_{v \Vdash \psi(x,Y)} \pi_v \\
&= \sum_{v \Vdash \psi(x,Y) \wedge x} \pi_v \quad + \sum_{v \Vdash \psi(x,Y) \wedge \neg x} \pi_v \\
&= 0 \quad + \sum_{v \Vdash \psi(x,Y) \wedge \neg x} \pi_v \\
&= \sum_{v \Vdash \psi(0,Y)} \pi_v \\
&= \mathsf{Pr}(\psi(0, Y))) \qquad .
\end{aligned}$$

The second case follows analogously. □

Proposition 4.1. *Given a formula $\varphi(X, K, R)$ the* GGenPSAT *problem $\mathsf{PM}(\varphi, \mathbb{P})$ is unsatisfiable iff $\varphi(X, K, R)$ is perfectly masked under $\mathbb{P}$.*

Proof. Assume that the $\mathsf{PM}(\varphi, \mathbb{P})$ is satisfiable and let π be the probability distribution over the set of valuations $\mathcal{V}$ on X, K, K', R that satisfies the constraints in $\mathsf{PM}(\varphi, \mathbb{P})$. Notice that since π satisfies $\mathsf{prop}(X)$ and $\mathsf{prop}(K \cup K')$, we apply Lemma 4.1 and conclude that there is an assignment $\mathbf{X}, \mathbf{K}, \mathbf{K'}$ for the variables in X, K, K' such that $\mathsf{Pr}(\varphi(X, K, R)) = \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R))$ and $\mathsf{Pr}(\varphi(X, K', R)) = \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K'}, R))$. Therefore,

$$\pi \Vdash \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R)) \neq \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K'}, R)) \ .$$

This, on the other hand, means that

$$\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} \pi_v \ .$$

Since each valuation is only free for the variables in R, and $\pi \Vdash \mathcal{R}(\mathbb{P})$

$$\begin{aligned}
&\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K}',R)} \pi_v \\
\Leftrightarrow &\sum_{\mathbf{R} \in \mathsf{dom}(R)} \pi_{\bar{v}_\mathbf{R}} \cdot \bar{v}_\mathbf{R}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \sum_{\mathbf{R} \in \mathsf{dom}(R)} \pi_{\bar{v}_\mathbf{R}} \cdot \bar{v}_\mathbf{R}(\varphi(\mathbf{X},\mathbf{K}',R)) \\
\Leftrightarrow &\sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_\mathbf{R}(\varphi(\mathbf{X},\mathbf{K},R)) \neq \sum_{\mathbf{R} \in \mathsf{dom}(R)} \mathsf{P}(\mathbb{P} = \mathbf{R}) \cdot \bar{v}_\mathbf{R}(\varphi(\mathbf{X},\mathbf{K}',R)) \\
&\Leftrightarrow \mathsf{P}(D_{\mathbb{P},\varphi}(\mathbf{X},\mathbf{K}) = 1) \neq \mathsf{P}(D_{\mathbb{P},\varphi}(\mathbf{X},\mathbf{K}') = 1)
\end{aligned}$$

and so φ is not perfectly masked under $\mathbb{P}$ according to Definition 4.3.

For the direct implication, observe that assuming the unsatisfiability of $\mathsf{PM}(\varphi, \mathbb{P})$, using Lemma 4.1, all possible instantiations would need to satisfy $\mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K},R)) = \mathsf{Pr}(\varphi(\mathbf{X},\mathbf{K}',R))$ and so, φ would be perfectly masked under $\mathbb{P}$. □

Provided this formulation for deciding if a formula is perfectly masked under $\mathbb{P}$ in GGenPSAT, the computational complexity of this problem becomes, in a way, parametrized by the size of the set of formulas that defines the probability distribution of the random variables. We recall that GGenPSAT is NP-complete [CCM17a], and thus, if both the number of formulas in $\mathcal{R}(\mathbb{P})$ and their size are polynomially bounded on the size of the original formula φ, the problem of deciding if a formula is perfectly masked, which corresponds to determining the unsatisfiability of $\mathsf{PM}(\varphi, \mathbb{P})$, lies in co-NP.

Proposition 4.2. *Let $\varphi(X, K, R)$ be a Boolean formula and $\mathbb{P}$ a probability distribution on R. If $|\mathcal{R}(\mathbb{P})| = \mathsf{poly}(|\varphi|)$, then the problem of deciding if φ is perfectly masked under $\mathbb{P}$ is in* co-NP.

4.4 Characterizing attackers with side-channel capabilities

In this section, we formally characterize four types of attackers with side-channel capabilities and study the computational problem of deciding if a Boo-lean formula is perfectly masked against each different type of attacker. The scenario comprises a Boolean formula which contains plaintext variables X, secret key variables K and random variables R which are independently sampled according to a $\mathsf{Bernoulli}(\frac{1}{2})$ probability distribution. The goal of the attacker is to be able to distinguish when different keys are being used. We will model four different types of attackers:

1. Passive attacker is only able to observe the result of the computation nodes.

2. Variable-dependency attacker is able to change the dependence and independence of the random masks being used in the computation node.

3. Fault-injection attacker is able to alter the probabilities of each random variable, making them biased or even deterministic.

4. General attacker has the power of a variable-dependency attacker as well as a fault-injection attacker.

4.4.1 Perfect Masking against a Passive Attacker

In the case that the random masks are independent and distributed according to a Bernoulli($\frac{1}{2}$) probability distribution, this problem is easily reducible to counting the number of satisfying assignments of φ depending on the value of the secret key. If this value differs, the formula is not perfectly masked. In [EWS14], the perfect masking problem is solved by a reduction to the satisfiability of a formula in SMT that states that the number of satisfying assignments of φ is independent of the secret parameters. Specifically, the authors define an equivalent formulation for the perfect masking problem as follows: determine if a formula $\varphi(X, K, R)$ is perfectly masked if for all instances of the variables in X, K, $\mathbf{X} \in \mathsf{dom}(X)$ and $\mathbf{K}, \mathbf{K}' \in \mathsf{dom}(K)$

$$\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R)) = \#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}', R)) \ ,$$

where for fixed values of $\mathbf{X}, \mathbf{K}$, $\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R))$ is the number of assignments over $r \in R$ in which the Boolean formula is satisfiable. If this formula is unsatisfiable, it means that there is a plaintext $\mathbf{X}$ and two keys $\mathbf{K}, \mathbf{K}'$ such that the probability distribution of the Boolean formula $\varphi(\mathbf{X}, \mathbf{K}, R)$ differs from $\varphi(\mathbf{X}, \mathbf{K}', R)$. This means that information about the secret key is being leaked and thus the computation is not perfectly masked. We will refer to this as the *perfect masking by counting* problem. Implementations wise, [EWS14] solves $\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}', R))$ by encoding it in an exponentially sized SMT formula on the number of variables in R.

We now describe how to model a passive attacker in the GGenPSAT framework. In this case, the random masks are independent random variables sampled according to a Bernoulli($\frac{1}{2}$) probability distribution, and so, the set $\mathcal{R}(\mathbb{P})$ is composed of $\mathsf{Pr}(r) = \frac{1}{2}$ for all $r \in R$. Furthermore, we need to specify that all these random variables are mutually independent, i.e., $\mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}}$ for any $S \in \wp_{\geq 2}(R)$.

Example 4.3. *For a specific example, consider* $\varphi(\{\}, \{k\}, \{r_1, r_2\}) = k \oplus (r_1 \oplus r_2)$. *Thus, the problem of deciding whether this computation is perfectly masked, assuming the random masks are mutually independent and sampled according to a* Bernoulli($\frac{1}{2}$) *probability distribution, rests on deciding the satisfiability of the*

GGenPSAT *problem*

$$\begin{cases} \mathsf{Pr}(k \oplus (r_1 \oplus r_2)) \neq \mathsf{Pr}(k' \oplus (r_1 \oplus r_2)) \\ \mathsf{prop}(k) \wedge \mathsf{prop}(k') \\ \mathsf{Pr}(r_i) = \frac{1}{2} \qquad \textit{for } i = 1, 2 \\ \mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{4} \end{cases}$$

which is unsatisfiable as we have seen in Subsection 3.6.2. This means that indeed, this formula is secure under independent random masks sampled according to a Bernoulli($\frac{1}{2}$) *probability distribution.*

In the general case, deciding if a Boolean formula $\varphi(X, K, R)$ is perfectly masked against a passive attacker rests on determining the satisfiability of the following GGenPSAT problem $\mathsf{PM}(\varphi, \mathsf{Passive})$:

$$\mathsf{PM}(\varphi, \mathsf{Passive}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} \qquad \text{for } S \in \wp_{\geq 2}(R) \end{cases}$$

In the next proposition, we show that this formulation of perfect masking is equivalent to the one introduced by Wang et al. [EWS14] – the perfect masking by counting problem. To do this, we first need to show that

$$\begin{cases} \mathsf{Pr}(r) = \frac{1}{2} & \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \text{for } S \in \wp_{\geq 2}(R) \end{cases}$$

defines the joint distribution of $|R|$ independent Bernoulli($\frac{1}{2}$) random variables.

Lemma 4.2. *If a probability distribution π over valuations on a set R satisfies*

$$\begin{cases} \mathsf{Pr}(r) = \frac{1}{2} & \textit{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \textit{for } S \in \wp_{\geq 2}(R) \end{cases}$$

then π is the joint probability distribution of $|R|$ independent Bernoulli($\frac{1}{2}$).

Proof. This is a restatement from a result in [Teu90] by noting that these correspond to the cross-moments of the joint random variables and that these $2^{|R|}$ parameters characterize precisely this probability distribution. An alternative formulation of this would be

$$\mathsf{Pr}(\bigwedge_{r \in S} r \wedge \bigwedge_{r \in R \setminus S} \neg r) = \frac{1}{2^{|R|}} \qquad \text{for } S \in \wp(R) \ ,$$

which explicitly defines the full joint probability distribution and therefore fully characterizes the distribution. □

Proposition 4.3. *Given a formula $\varphi(X, K, R)$ the* GGenPSAT *problem* $\mathsf{PM}(\varphi, \mathsf{Passive})$ *is unsatisfiable iff $\varphi(X, K, R)$ is perfectly masked according to the perfect masking by counting problem.*

Proof. Assume that the $\mathsf{PM}(\varphi, \mathsf{Passive})$ is satisfiable and let π be the probability distribution over the set of valuations $\mathcal{V}$ on X, K, K', R that satisfies the constraints in $\mathsf{PM}(\varphi, \mathsf{Passive})$. Notice that since π satisfies $\mathsf{prop}(X)$ and $\mathsf{prop}(K \cup K')$, we apply Lemma 4.1 and conclude that there is an assignment $\mathbf{X}, \mathbf{K}, \mathbf{K'}$ for the variables in X, K, K' such that $\mathsf{Pr}(\varphi(X, K, R)) = \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R))$ and $\mathsf{Pr}(\varphi(X, K', R)) = \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K'}, R))$. Therefore,

$$\pi \Vdash \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R)) \neq \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K'}, R)) \ .$$

This, on the other hand, means that

$$\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \pi_v \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} \pi_v \ .$$

These only depend on the variables in R and by Lemma 4.2 we know that π must be sampled according to the joint probability of $|R|$ independent $\mathsf{Bernoulli}(\frac{1}{2})$ variables on each valuation. Thus,

$$\begin{gathered}
\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} \frac{1}{2^{|R|}} \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} \frac{1}{2^{|R|}} \\
\frac{1}{2^{|R|}} \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} 1 \neq \frac{1}{2^{|R|}} \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} 1 \\
\sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K},R)} 1 \neq \sum_{v \Vdash \varphi(\mathbf{X},\mathbf{K'},R)} 1 \\
\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R)) \neq \#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K'}, R)) \ ,
\end{gathered}$$

which shows that indeed φ is not perfectly masked by counting.

For the direct implication, observe that assuming the unsatisfiability of $\mathsf{PM}(\varphi, \mathsf{Passive})$, using Lemma 4.1, all possible instantiations would need to satisfy

$$\mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K}, R)) = \mathsf{Pr}(\varphi(\mathbf{X}, \mathbf{K'}, R))$$

and so, Lemma 4.2 would imply,

$$\#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K}, R)) = \#\mathsf{SAT}(\varphi(\mathbf{X}, \mathbf{K'}, R)) \ .$$

□

The number of probabilistic formulas necessary to specify the probability distribution of the random masks is as many as[1] the number of subsets of R, i.e.,

[1] in fact we only require $2^{|R|} - 1$ of these statements given that they all must add up to one.

$2^{|R|}$. This amounts to an exponential sized GGenPSAT instance which is not an improvement regarding the encoding in SMT of [EWS14]. However, as we will see in the next sections, the expressiveness of GGenPSAT allows us to model situations in which the attackers actively interfere with the system. Such situations are intrinsically probabilistic and cannot be easily modelled in SMT or with counting problems.

4.4.2 Perfect Masking against a Variable-dependency Attacker

In this section, we study the problem of deciding the perfect masking of a Boolean formula against an attacker which is capable of manipulating the dependency of the different random variables used as masks. These attackers are able to make variables depend on each other, where originally they were independent. In this way, we can construct a family of problems which consist in deciding perfect masking against each different variable-dependency attacker.

In the GGenPSAT formalism, this is done by considering the problem unconstrained by the independence requirements. Consider the following GGenPSAT problem parametrized by the Boolean formula φ as well as the set $\mathcal{A}_{vd} \subseteq \wp_{\geq 2}(R)$, which characterizes the power of the attacker by defining the sets of variables for which the attacker is able to interfere with

$$\mathsf{PM}(\varphi, \mathcal{A}_{vd}) : \begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) & \\ \mathsf{prop}(X) & \\ \mathsf{prop}(K \cup K') & \\ \mathsf{Pr}(r) = \frac{1}{2} & \text{for } r \in R \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} & \text{for all } S \in \wp_{\geq 2}(R) \setminus \mathcal{A}_{vd} \end{cases}$$

In this setting, an attacker is able to manipulate any variable dependency specified by the set $\mathcal{A}_{vd}$. Also note that a passive attacker is characterized by $\mathcal{A}_{vd} = \emptyset$.

Definition 4.5. *A Boolean formula is perfectly masked against a variable-dependency attacker $\mathcal{A}_{vd}$ if the* GGenPSAT *problem $\mathsf{PM}(\varphi, \mathcal{A}_{vd})$ is unsatisfiable.*

By not imposing any independence restriction, $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$, we model the strongest possible variable-dependency attacker. Furthermore, as noted previously in Proposition 4.2, with $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$, we obtain $\mathcal{R}(\mathbb{P}) = \{\mathsf{Pr}(r) = \frac{1}{2} \mid r \in R\}$, which has linear size on the size of R, and so this problem becomes co-NP.

Proposition 4.4. *Given a Boolean formula φ and a variable-dependency attacker $\mathcal{A}_{vd} = \wp_{\geq 2}(R)$ the problem of deciding if φ is perfectly masked against $\mathcal{A}_{vd}$ is in* co-NP.

Proof. In this case, determining if φ is perfectly masked against a variable-dependency attacker corresponds to the unsatisfiability of the GGenPSAT problem $\mathsf{PM}(\varphi, \mathcal{A}_{vd})$:

$$\begin{cases} \mathsf{Pr}(\varphi(X,K,R)) \neq \mathsf{Pr}(\varphi(X,K',R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad\qquad \text{for } r \in R \end{cases}$$

which has linear size in the size of φ. This shows the problem is in co-NP since the GGenPSAT problem is in NP. □

Proposition 4.5. *Consider a Boolean formula φ and a variable-dependency attacker $\mathcal{A}_{vd}$ such that $|\mathcal{P}_{\geq 2}(R) \setminus \mathcal{A}_{vd}| = \mathsf{poly}(|\varphi|)$. The problem of deciding if φ is perfectly masked against $\mathcal{A}_{vd}$ is in* co-NP.

Proof. In this case, we have a GGenPSAT problem with $|\mathcal{P}_{\geq 2}(R) \setminus \mathcal{A}_{vd}| + |R| + 2|K| + |X| + 1$ probabilistic formulas. By hypothesis, this set has polynomial size on $|\varphi|$ and so this problem is in co-NP. □

4.4.3 Perfect Masking against a Fault-injection Attacker

An attacker capable of fault-injection is characterized by its ability to control program variables and program flow by inserting hardware flaws. In our setting, we can model this type of behaviour by describing an attacker that is able to manipulate the random variables of the system. This manipulation can either mean that the attacker can skew $\mathsf{Bernoulli}(\frac{1}{2})$ distribution of a random mask r, and, for instance, impose that $\mathsf{Pr}(r) \geq b$, or even to deterministically control a variable, i.e., impose that $\mathsf{Pr}(r) = 0 \vee \mathsf{Pr}(r) = 1$ or even that $\mathsf{Pr}(r) = 1$.

We characterize each fault-injection attacker $\mathcal{A}_{fi}$ by the set of random variables it manipulates, also called the insecure random variables and denoted by $\mathsf{InsR} \subseteq R$. Furthermore, this set is divided in two disjoint sets $\mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns}$: the set of fully controlled variables FIns and the set of partially controlled variables, PIns. By a fully controlled variable r, we mean the attacker can impose any probabilistic assertion on the system regarding that variable. Since the modelling is done by specifying what the attacker must adhere to, no GGenPSAT formula is imposed on the FIns variables.

On the other hand, an attacker can only influence a partially controlled variable in a specific manner determined by a set of GGenPSAT formulas involving only variables in PIns, $\Delta(\mathsf{PIns}) \subseteq \mathsf{Prob}(\mathsf{PIns})$ that explicitly defines the control that the attacker has over these variables. For instance an attacker might be able to skew the probability distribution of r, but not know exactly how it changed: $\mathsf{Pr}(r) < 0.2 \vee \mathsf{Pr}(r) > 0.8$.

In summary, $\mathcal{A}_{fi} = \langle \mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns},\ \Delta(\mathsf{PIns}) \rangle$.

This way, consider the GGenPSAT problem parametrized by a Boolean formula φ as well as an attacker $\mathcal{A}_{fi} = \langle \mathsf{InsR} = \mathsf{FIns} \cup \mathsf{PIns},\ \Delta(\mathsf{PIns}) \rangle$,

$$\mathsf{PM}(\varphi, \mathcal{A}_{fi}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \psi \qquad \text{where } \psi \in \Delta(\mathsf{PIns}) \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad \text{for } r \in R \setminus \mathsf{InsR} \\ \mathsf{Pr}(\bigwedge_{r \in S} r) = \frac{1}{2^{|S|}} \qquad \text{for } S \in \wp_{\geq 2}(R \setminus \mathsf{InsR}) \end{cases}$$

If this problem is unsatisfiable, this means that it is not possible to distinguish the behaviour of the Boolean formula φ when different secret keys are being used, even when manipulating some of the masking variables. In other words, this means that the formula is perfectly-masked against an attacker with fault-injection capabilities.

Definition 4.6. *A formula φ is perfectly masked against a fault-injection attacker $\mathcal{A}_{fi}$ if the* GGenPSAT *problem $\mathsf{PM}(\varphi, \mathcal{A}_{fi})$ is unsatisfiable.*

4.4.4 Perfect Masking against a general attacker

In this section, we consider a general attacker which can both inject faults, as well change the dependency of the random variables.

An attacker is characterized by their ability to

- manipulate probabilities of the random masks in the set of insecure random variables InsR. From this set, the attacker can either have full control of the variable or only partial control, as well as specify their interaction with other random variables. This is specified by a set of GGenPSAT formulas involving variables in R, $\Delta(R) \subseteq \mathsf{Prob}(R)$, that explicitly defines the control that the attacker has over these variables.

Thus, given a Boolean formula φ and an attacker

$$\mathcal{A} = \langle \mathsf{InsR},\ \Delta(R) \rangle,$$

denote by $\mathsf{PM}(\varphi, \mathcal{A})$ the GGenPSAT problem defined by the following formulas

$$\mathsf{PM}(\varphi, \mathcal{A}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \psi \qquad \text{where } \psi \in \Delta(R) \\ \mathsf{Pr}(r) = \frac{1}{2} \qquad \text{for } r \in R \setminus \mathsf{InsR} \end{cases}$$

Definition 4.7. *A Boolean formula φ is perfectly masked against a general attacker $\mathcal{A}$ if the* GGenPSAT *problem* $\mathsf{PM}(\varphi, \mathcal{A})$ *is unsatisfiable.*

Example 4.4. *We now present a masking scheme of a secret k which aims to be secure even when an attacker has control over the probability p of random masks. We make use of a variation of a coin debiasing trick due to von Neumann [vN51], in which given a biased coin* $\mathsf{Bernoulli}(p)$ *with $p \neq 0, 1$, we produce a* $\mathsf{Bernoulli}(\frac{1}{2})$ *coin. The method is as follows: flip the coin twice; if the result is HT or TH, return H or T, respectively; otherwise, start the process again. Our masking scheme relies on 3 random masks, r_1, r_2, r_3 in which all are independent, r_1, r_2 are distributed as* $\mathsf{Bernoulli}(p)$ *and r_3 is distributed as* $\mathsf{Bernoulli}(\frac{1}{2})$.

The motivation is that there are two randomness sources, and one of them is potentially controlled by an attacker. However, due to debiasing, we protect against the control on the $\mathsf{Bernoulli}(p)$ *randomness source. This way, an attacker would have to control both randomness sources to actually be able to distinguish different keys. In other words, r_1, r_2 can be viewed as two coin flips of the* $\mathsf{Bernoulli}(p)$ *random variable and r_3 a fair coin flip. The masking scheme* xor*'s the key k with one of two values:*

- *either $r_1 \oplus r_2$ holds, and so their values are different, which means we are in the debiasing conditions, and we return r_1, or*
- *$r_1 \oplus r_2$ does not hold and so we have to resort to the fail-safe coin r_3.*

This can be expressed as the masking scheme φ given by

$$\varphi \triangleq k \oplus ((r_1 \oplus r_2 \rightarrow r_1) \wedge ((\neg(r_1 \oplus r_2)) \rightarrow r_3)) \ .$$

We conjecture that indeed this masking scheme is secure, for the case $p = \frac{1}{3}$, as we have not been able to find a satisfiable assignment of the following set of formulas in the developed GGenPSAT *solver, [CMC16b], running for 300 hours:*

$$\begin{cases} \mathsf{Pr}(\varphi(k, r_1, r_2, r_3)) \neq \mathsf{Pr}(\varphi(k', r_1, r_2, r_3)) \\ \mathsf{prop}(\{k, k'\}) \\ \mathsf{Pr}(r_1 \wedge r_2 \wedge r_3) = \frac{1}{18} \\ \mathsf{Pr}(r_1 \wedge r_2) = \frac{1}{9} \\ \mathsf{Pr}(r_1 \wedge r_3) = \frac{1}{6} \\ \mathsf{Pr}(r_2 \wedge r_3) = \frac{1}{6} \\ \mathsf{Pr}(r_1) = \frac{1}{3} \\ \mathsf{Pr}(r_2) = \frac{1}{3} \\ \mathsf{Pr}(r_3) = \frac{1}{2} \end{cases}$$

4.5 Conclusions and Future Work

In this chapter we studied applications of formal-verification, namely satisfiability problems, in the area of cryptography, specifically on side-channel analysis and their mitigation techniques. A common, and usually efficient technique that mitigates power-related side-channel attacks is Boolean masking: these techniques work by applying a random (and unknown) mask to a computation step, in order to *mask* the secret keys and other values that are used in said computation. However, if these masks are not designed in a proper way, even when they are correctly applied, they could leak information regarding the secrets they are designed to hide.

Here, we modelled the problem of deciding if a Boolean formula is perfectly masked in the probabilistic formalism. Furthermore, we generalized this scenario to encompass an active attacker interfering with the Boolean masks, by changing dependency between random variables, or actually changing their probability distributions. Remarkably, we found that solving the problem of deciding if a Boolean formula is perfectly masked is (in terms of computational complexity) easier to solve in the presence of more powerful attackers, lowering the complexity of this problem to co-NP in some cases. In practical terms, this shows that it is easier to decide if a formula is perfectly masked when powerful attackers are considered and we are actually gaining more security guarantees by solving an easier problem. However, naturally, the set of formulas which are secure against active attackers is much smaller than when a passive adversary is considered.

We believe that now that there is a formal probabilistic model of perfect masking and some classes of active attackers, several avenues for further research have opened. On one hand, the formal verification of modern masking schemes used in current technologies, as has been done in [EWS14], can now be done considering more powerful attackers. On the other hand, one can now model other classes of active attackers that, for instance introduce faulty gates in the system. This type of situation can easily be modelled as we showed in Subsection 3.6.1.

One can also consider a variant problem of perfect masking in which we allow for some fluctuations between the probability of masks with different keys. Thus, instead of considering the problem of finding if

$$\mathsf{PM}(\varphi, \mathbb{P}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathcal{R}(\mathbb{P}) \end{cases}$$

is unsatisfiable, we could choose a tolerance parameter $\varepsilon \in [0, 1]$ and consider the problem

$$\mathsf{PM}_{\varepsilon}(\varphi, \mathbb{P}) : \begin{cases} \mathsf{Pr}(\varphi(X, K, R)) \neq_{\varepsilon} \mathsf{Pr}(\varphi(X, K', R)) \\ \mathsf{prop}(X) \\ \mathsf{prop}(K \cup K') \\ \mathcal{R}(\mathbb{P}) \end{cases}$$

where $a \neq_{\varepsilon} b$ denotes that $(a > b + \varepsilon) \vee (a < b - \varepsilon)$. In this scenario, the two probabilities must differ of at least ε in case the formula is not perfectly masked. This allows to look for keys $\mathbf{K}, \mathbf{K'}$ which make the probabilities easily distinguishable, or above the tolerance of some measurement equipment.

Chapter 5

Conclusion and Future Work

In this dissertation, we aimed to study two areas that are essential to the application of formal methods to real-world systems. In particular, in Chapter 1, we studied in which conditions satisfiability procedures can be combined, in order to obtain a procedure which decides the satisfiability of formulas in a more complex theory. In fact, we generalized the notion of shiny theories to the many-sorted case and proved that this class coincides with the class of strongly polite theories. This allowed us to develop two combination methods for a many-sorted shiny theory – one using this equivalence, and a direct method.

Thus, after showing that shiny theories are equivalent to strongly polite theories, a question immediately arises: is there a class of theories, different from these, that also has a combination procedure with an arbitrary theory?

In the second part of this work, we focused on a different aspect of real-world systems which is on how to deal, using formal methods, with imprecise, uncertain and probabilistic statements. We noticed that despite several probabilistic logics have been proposed and studied, there is a lack of available tools and constructible satisfiability procedures for them. Specifically, in Chapter 2, we studied a fragment of the probabilistic logic of Fagin et al., which is able to express assertions which are linear combinations of probabilistic formulas. To develop a tool that solves the satisfiability procedure for this logic, we found a polynomial reduction of this problem to Mixed Integer Programming and showed its correctness. Having the reduction in hands, an open-source solver was developed and tested with reasonably sized instances of the problem. Furthermore, this allowed us to study phase transition behaviours for this problem and identified several parameter ratios for the occurrence of these phase transitions.

As described during this work, we found these phase transition parameters experimentally with our developed solvers. However, as future work avenues, we believe that formally deriving an interval in which the critical phase transition parameter belongs can be done using similar arguments as was previously done

for 3SAT, [BHvM09].

After developing the satisfiability procedure for a fragment of Fagin et al. probabilistic logic, it was only natural to study how to develop a satisfiability procedure for the full language of the logic. This was the work done in Chapter 3 which followed a similar line of thought as the previous chapter: we started by finding a natural reduction for this problem, this time to the satisfiability problem of the theory of quantifier-free linear integer and real arithmetic. After proving its correctness, we implemented the solver making use of off-the-shelf SMT solvers. Similarly to the previous chapter, we studied the phase transition behaviour of this problem.

We then showcased how this probabilistic formalism can indeed be used to model interesting and useful problems. In Section 3.6 we presented simple instances of this, namely to hardware verification and side-channel analysis of cryptographic programs. Besides the description of the problem, we modelled the instances in the developed tool and provided the code for the examples. Then, in Chapter 4, we took a deeper look into the detection of side-channel attacks in masked cryptographic programs. There, we modelled the problem of deciding whether a Boolean formula is perfectly masked in the probabilistic formalism. Furthermore, we generalized this scenario to encompass an active attacker interfering with the Boolean masks, by changing dependency between random variables, or actually changing their probability distributions. Remarkably, we found that solving the problem of deciding if a Boolean formula is perfectly masked is (in terms of computational complexity) easier to solve in the presence of more powerful attackers, lowering the complexity of this problem to co-NP in some cases.

Several problems are worth mentioning as future work. On one hand, the formal verification of modern masking schemes used in current technologies, as has been done in [EWS14], can now be done considering more powerful attackers. On the other hand, one can now model other classes of active attackers that, for instance introduce faulty gates in the system.

Bibliography

[AARR03] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side–Channel(s). In *Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop*, pages 29–45. Springer Berlin Heidelberg, 2003.

[Ada96] E. W. Adams. *A primer of probability logic*. Stanford, CSLI Publications, 1996.

[AF11] C. Areces and P. Fontaine. Combining Theories: The Ackerman and Guarded Fragments. In *Frontiers of Combining Systems: 8th International Symposium, FroCoS*, pages 40–54. Springer Berlin Heidelberg, 2011.

[AG97] 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.

[Bal10] P. Baltazar. *Probabilization of Logic Systems*. PhD thesis, IST - Technical University of Lisbon, Portugal, 2010.

[BBD+15] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, and P. Strub. Verified proofs of higher-order masking. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–485, 2015.

[BCF15] G. Bona, F. G. Cozman, and M. Finger. Generalized probabilistic satisfiability through integer programming. *Journal of the Brazilian Computer Society*, 21(1):1–14, 2015.

[BDL97] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology - EUROCRYPT'97*, pages 37–51. Springer, 1997.

[BDL01] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of cryptology*, 14(2):101–119, 2001.

[BFT16] C. Barrett, P. Fontaine, and C. Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). `www.SMT-LIB.org`, 2016.

[BHvM09] A. Biere, M. Heule, and H. van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[BK00] F. Bacchus and F. Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1-2):123–191, 2000.

[Boo53] G. Boole. *Investigation of The Laws of Thought On Which Are Founded the Mathematical Theories of Logic and Probabilities*. Dover, 1853.

[BRNI13] A. G. Bayrak, F. Regazzoni, D. Novo, and P. Ienne. Sleuth: Automated verification of software power analysis countermeasures. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 293–310. Springer, 2013.

[Bur69] J. P. Burgess. Probability logic. *The Journal of Symbolic Logic*, 34(2):264–274, 1969.

[Car50] R. Carnap. *Logical foundations of probability*. The University of Chicago Press, 1950.

[CCM17a] C. Caleiro, F. Casal, and A. Mordido. Classical generalized probabilistic satisfiability. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 908–914, 2017.

[CCM17b] C. Caleiro, F. Casal, and A. Mordido. Generalized probabilistic satisfiability. *Electronic Notes in Theoretical Computer Science*, 332(Supplement C):39 – 56, 2017.

[CCM17c] C. Caleiro, F. Casal, and A. Mordido. Generalized probabilistic satisfiability and applications to modelling attackers with side-channel capabilities. *SQIG - Instituto de Telecomunicações and IST - U Lisboa, Portugal. Submitted for publication. Available online at http://sqig.math.ist.utl.pt/pub/CaleiroC/17-MCC-probmasking.pdf*, 2017.

[CFR14a] P. Chocron, P. Fontaine, and C. Ringeissen. A gentle non-disjoint combination of satisfiability procedures. In *International Joint Conference on Automated Reasoning*, pages 122–136. Springer, 2014.

[CFR14b] P. Chocron, P. Fontaine, and C. Ringeissen. A gentle non-disjoint combination of satisfiability procedures. In *Automated Reasoning:*

7th International Joint Conference, IJCAR, pages 122–136. Springer International Publishing, 2014.

[CGP99] E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT press, 1999.

[CH99] V. Chandru and J. Hooker. *Optimization methods for logical inference*. John Wiley and sons, Inc, 1999.

[CHSvdM16] C. Cremers, M. Horvat, S. Scott, and T. van der Merwe. Automated analysis and verification of TLS 1.3: 0-RTT, resumption and delayed authentication. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 470–485. IEEE, 2016.

[Chv83] V. Chvátal. *Linear programming*. Macmillan, 1983.

[CI13] F. G. Cozman and L. F. Ianni. Probabilistic satisfiability and coherence checking through integer programming. In *ECSQARU 2013. Proceedings*, pages 145–156. Springer Berlin Heidelberg, 2013.

[CJRR99] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology — CRYPTO' 99: 19th Annual International Cryptology Conference*, pages 398–412. Springer Berlin Heidelberg, 1999.

[CKT91] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–340, 1991.

[CLO07] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., 2007.

[CMC16a] F. Casal, A. Mordido, and C. Caleiro. GenPSAT solver, 2016. Available online at https://github.com/fcasal/genpsat.git.

[CMC16b] F. Casal, A. Mordido, and C. Caleiro. GGenPSAT solver, 2016. Available online at https://github.com/fcasal/ggenpsat.git.

[Coo71] 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.

[CR13] F. Casal and J. Rasga. Revisiting the equivalence of shininess and politeness. In *Logic for Programming, Artificial Intelligence, and Reasoning: 19th International Conference, LPAR-19*, pages 198–212. Springer Berlin Heidelberg, 2013.

[CR17] F. Casal and J. Rasga. Many-sorted equivalence of shiny and strongly polite theories. *Journal of Automated Reasoning*, 2017.

[CW96] E. M. Clarke and J. M. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4):626–643, 1996.

[Dic13] L. E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

[DKM+11] C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin. Sorting and selection in posets. *SIAM J. Comput.*, 40(3):597–622, 2011.

[DKR94] E. Domenjoud, F. Klay, and C. Ringeissen. Combination techniques for non-disjoint equational theories. In *International Conference on Automated Deduction*, pages 267–281. Springer, 1994.

[DKW08] V. D'silva, D. Kroening, and G. Weissenbacher. A survey of automated techniques for formal software verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1165–1178, 2008.

[DLV03] P. Dusart, G. Letourneux, and O. Vivolo. Differential fault analysis on AES. In *Applied Cryptography and Network Security*, pages 293–306. Springer, 2003.

[DMB08] L. De Moura and N. Bjørner. Z3: An efficient SMT solver. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, 2008.

[DMB11] L. De Moura and N. Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.

[Dut14] B. Dutertre. Yices 2.2. In *Computer-Aided Verification (CAV'2014)*, volume 8559 of *Lecture Notes in Computer Science*, pages 737–744. Springer, 2014.

[End01] H. Enderton. *A mathematical introduction to logic*. Academic press, 2001.

[EW14] H. Eldib and C. Wang. Synthesis of masking countermeasures against side channel attacks. In *International Conference on Computer Aided Verification*, pages 114–130. Springer, 2014.

[EWS14] H. Eldib, C. Wang, and P. Schaumont. Formal verification of software countermeasures against side-channel attacks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 24(2):11, 2014.

[FB11] M. Finger and G. Bona. Probabilistic satisfiability: Logic-based algorithms and phase transition. In *IJCAI*, pages 528–533. IJCAI/AAAI, 2011.

[FB15] M. Finger and G. Bona. Probabilistic satisfiability: algorithms with the presence and absence of a phase transition. *Annals of Mathematics and Artificial Intelligence*, 75(3):351–389, 2015.

[FHM90] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87(1-2):78–128, 1990.

[Fon09] P. Fontaine. Combinations of theories for decidable fragments of first-order logic. In *Frontiers of Combining Systems: 7th International Symposium, FroCoS*, pages 263–278. Springer Berlin Heidelberg, 2009.

[GHR98] D. M. Gabbay, C. J. Hogger, and J. A. Robinson. *Handbook of Logic in Artificial Intelligence and Logic Programming: Volume 5: Logic Programming*. Clarendon Press, 1998.

[GKP88] G. Georgakopoulos, D. Kavvadias, and C. H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4(1):1 – 11, 1988.

[GMO01] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems — CHES 2001: Third International Workshop*, pages 251–261. Springer Berlin Heidelberg, 2001.

[GO15] I. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[GPPT16] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer. ECDH key-extraction via low-bandwidth electromagnetic attacks on pcs. In *Cryptographers' Track at the RSA Conference*, pages 219–235. Springer, 2016.

[Gup92] A. Gupta. Formal hardware verification methods: A survey. In *Computer-Aided Verification*, pages 5–92. Springer, 1992.

[GW94] I. P. Gent and T. Walsh. *The hardest random SAT problems*. Springer, 1994.

[Hil70] D. Hilbert. *Über die Theorie der algebraischen Formen*, pages 199–257. Springer Berlin Heidelberg, 1970.

[HS13] M. Hutter and J.-M. Schmidt. The temperature side channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications*, pages 219–235. Springer, 2013.

[HT73] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, 1973.

[ISW03] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In *Annual International Cryptology Conference*, pages 463–481. Springer, 2003.

[JB10a] D. Jovanović and C. Barrett. Polite theories revisited. In *Logic for Programming, Artificial Intelligence, and Reasoning: 17th International Conference, LPAR-17*, pages 402–416. Springer Berlin Heidelberg, 2010.

[JB10b] D. Jovanović and C. Barrett. Polite theories revisited – extended version. Technical Report TR2010-922, Department of Computer Science, New York University, January 2010.

[KEH+09] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, et al. seL4: Formal verification of an OS kernel. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pages 207–220. ACM, 2009.

[Kin14] T. King. *Effective Algorithms for the Satisfiability of Quantifier-Free Formulas Over Linear Real and Integer Arithmetic*. PhD thesis, Courant Institute of Mathematical Sciences New York, 2014.

[KJJ99] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397. Springer-Verlag, 1999.

[Koc96] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology — CRYPTO '96: 16th Annual International Cryptology Conference*, pages 104–113. Springer Berlin Heidelberg, 1996.

[LLK07] A. Lamas-Linares and C. Kurtsiefer. Breaking a quantum key distribution system through a timing side channel. *Optics express*, 15(15):9388–9393, 2007.

[MC17] A. Mordido and C. Caleiro. Probabilistic logic over equations and domain restrictions. *accepted in Mathematical Structures in Computer Science*, 2017.

[Mea94] C. A. Meadows. Formal verification of cryptographic protocols: A survey. In *International Conference on the Theory and Application of Cryptology*, pages 133–150. Springer, 1994.

[MOP07] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., 2007.

[Mor17] A. Mordido. *A probabilistic logic over equations and domain restrictions*. PhD thesis, IST - Universidade de Lisboa, Portugal, 2017.

[MS95] S. P. Miller and M. Srivas. Formal verification of the AAMP5 microprocessor: A case study in the industrial use of formal methods. In *Workshop on Industrial-Strength Formal Specification Techniques*, pages 2–16. IEEE, 1995.

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

[NFSM+09] S. Nauerth, M. Fürst, T. Schmitt-Manderbach, H. Weier, and H. Weinfurter. Information leakage via side channels in freespace BB84 quantum cryptography. *New Journal of Physics*, 11(6):065001, 2009.

[Nil86] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28(1):71–88, 1986.

[NO79] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.

[Opp80] D. C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12(3):291 – 302, 1980.

[PR13] E. Prouff and M. Rivain. Masking against side-channel attacks: A formal security proof. In *Advances in Cryptology – EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 142–159. Springer Berlin Heidelberg, 2013.

[PS82] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1982.

[QS01] J.-J. Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA): Measures and counter-measures for smart cards. In *Smart Card Programming and Security: International Conference on Research in Smart Cards, E-smart*, pages 200–210. Springer Berlin Heidelberg, 2001.

[Ram17] C. Ramsay. TEMPEST attacks against AES. Technical Report White paper, Fox-IT, 2017.

[Ros09] J. Rosenhouse. *The Monty Hall problem: the remarkable story of Math's most contentious brain teaser*. Oxford University Press, 2009.

[RRZ05] S. Ranise, C. Ringeissen, and C. G. Zarba. Combining data structures with nonstably infinite theories using many-sorted logic. In *Frontiers of Combining Systems: 5th International Workshop, FroCoS*, pages 48–64. Springer Berlin Heidelberg, 2005.

[SSS00] M. Sheeran, S. Singh, and G. Stålmarck. Checking safety properties using induction and a SAT-solver. In *International conference on formal methods in computer-aided design*, pages 127–144. Springer, 2000.

[Teu90] J. L. Teugels. Some representations of the multivariate Bernoulli and binomial distributions. *Journal of Multivariate Analysis*, 32(2):256 – 268, 1990.

[TH96] C. Tinelli and M. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In *Frontiers of Combining Systems: First International Workshop*, pages 103–119. Springer Netherlands, 1996.

[TR03] C. Tinelli and C. Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 290(1):291–353, 2003.

[Tse68] G. S. Tseitin. On the complexity of derivations in the propositional calculus. *Studies in Mathematics and Mathematical Logic*, Part II:115–125, 1968.

[TZ05] C. Tinelli and C. G. Zarba. Combining nonstably infinite theories. *Journal of Automated Reasoning*, 34(3):209–238, 2005.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *Appl. Math Ser*, 35(12):36–38, 1951.

[WJ94] M. Wooldridge and N. R. Jennings. Agent theories, architectures, and languages: a survey. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 1–39. Springer, 1994.

[WJ95] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.

[Zar02] C. G. Zarba. A tableau calculus for combining non-disjoint theories. In *TABLEAUX*, volume 2, pages 315–329. Springer, 2002.