

UNIVERSITÄT BREMEN
FACHBEREICH 3 – MATHEMATIK/INFORMATIK

Dissertation

Reasoning in Many Dimensions: Uncertainty and Products of Modal Logics

eingereicht am 17.07.2014 von

Jean Christoph Jung, M.Sc.

geboren am 12.08.1984 in Bad Salzungen.

Kolloquium: 06.10.2014

Gutachter:

Prof. Dr. rer. nat. Carsten Lutz, Universität Bremen

Prof. Dr. Diego Calvanese, Freie Universität Bozen-Bolzano

Abstract

Description Logics (DLs) are a popular family of knowledge representation languages. They are fragments of first-order logic (FO) that combine high expressiveness with reasonable computational properties; in particular, most DLs are decidable. However, being based on first-order logic, they share also the shortcomings. One of these shortcomings is that DLs do not have built-in means to capture *uncertainty*, a feature that is commonly needed in many applications. This problem has been addressed in many different ways; one of the most recent proposals is the introduction of *Probabilistic Description Logics (ProbDLs)* which relate to *Probabilistic first-order logic (ProbFO)* in the same way as DLs relate to standard FO. In order to capture the uncertainty, ProbFO and thus ProbDLs adopt a possible world semantics. More specifically, a ProbDL or ProbFO knowledge base describes a family of distributions over possible worlds.

These logics constitute the scope of the first part of the thesis. We investigate the following settings:

- Reasoning in full ProbFO is highly undecidable and standard restrictions like the guarded fragment do not lead to decidability. We identify a fragment, *monodic* ProbFO, that shows several nice properties: the validity problem is recursively enumerable and decidability of FO fragments carries over to the corresponding monodic ProbFO fragment;
- In order to identify well-behaved, in best-case tractable ProbDLs, we study the complexity landscape for different fragments of Prob \mathcal{EL} ; amongst others, we are able to identify a tractable fragment.
- We then turn our attention to the recently popular reasoning problem of ontological query answering, but apply it to probabilistic data. More precisely, we define the framework of *ontology-based access to probabilistic data* and study the computational complexity therein. The main results here are *dichotomy theorems* between PTime and #P.

Probabilistic logics as described above can be viewed as instances of the framework of *many-dimensional logics*, one dimension being classical logic and the other being reasoning with probabilities. In the final part of the thesis, we remain in this framework and study the complexity of the satisfiability problem in the two-dimensional modal logic $\mathbf{K} \times \mathbf{K}$. Particularly, we are able to close a gap that has been open for more than ten years.

Zusammenfassung

Beschreibungslogiken (BLn) sind eine häufig betrachtete Sprachenfamilie zur Wissensrepräsentation. Sie sind Fragmente der Logik erster Stufe (FO), die gute Ausdruckstärke mit angemessenen Berechnungseigenschaften kombinieren. Da BLn auf der Semantik von FO basieren, erben sie deren Unzulänglichkeiten. Eine solche ist, dass weder FO noch BLn Mittel zur Repräsentation von *Unsicherheit* zur Verfügung stellen, was aber in vielen Anwendungen notwendig ist. Für dieses Problem wurde schon eine Vielzahl von Ansätzen vorgeschlagen; einer der aktuellsten sind *Probabilistische Beschreibungslogiken* (*ProbBLn*), die sich zu *Probabilistischer Logik erster Stufe* (*ProbFO*) so verhalten wie klassische BLn zu FO. Diesen probabilistischen Logiken unterliegt eine “possible world”-Semantik um die Unsicherheit zu erfassen: eine ProbBL- oder ProbFO-Wissensbasis beschreibt eine Menge von möglichen Verteilungen über possible worlds.

Diese Logiken bilden den Rahmen des ersten Teils der Arbeit, in welchem die folgenden Szenarien betrachtet werden:

- Es ist bekannt, dass Erfüllbarkeit in vollem ProbFO hochgradig unentscheidbar ist. Es wird das *monodische Fragment* von ProbFO eingeführt und gezeigt, dass es gute Eigenschaften aufweist. Zum Beispiel ist das Gültigkeitsproblem rekursiv aufzählbar und Entscheidbarkeit von FO-Fragmenten überträgt sich auf die entsprechenden Fragmente von monodischem ProbFO.
- Mit dem Ziel, ProbBLn zu finden, die effizientes Schlussfolgern zulassen, wird eine nahezu vollständige Analyse der Fragmente von Prob \mathcal{EL} durchgeführt.
- Ein in den letzten Jahren sehr gut untersuchtes Szenario im Zusammenhang mit BLn ist der Ontologie-basierte Datenzugriff. Da auch hier in vielen Anwendungen mit unsicheren Daten umgegangen werden muss, wird das Framework des *Ontologie-basierten Datenzugriffs auf probabilistische Daten* definiert und die Komplexität von Anfragebeantwortung darin untersucht. Die Hauptresultate in diesem Abschnitt sind *Dichotomietheoreme* zwischen PTime und #P.

Die beschriebenen probabilistischen Logiken können als Instanzen *mehr-dimensionaler Logiken* aufgefasst werden: eine Dimension ist hierbei die unterliegende klassische Logik und die andere das Rechnen mit Wahrscheinlichkeiten. Im letzten Teil der Arbeit werden weitere mehr-dimensionale Logiken untersucht. Das zentrale Resultat ist die inherente Nichtelementarität des Erfüllbarkeitsproblems für die Logik $\mathbf{K} \times \mathbf{K}$. Damit wird eine seit mehreren Jahren offene Frage beantwortet.

Acknowledgements

This thesis is the result of more than four years work as a PhD student. However, it would not have been possible to write it without the support of so many people.

First of all, I want to thank my advisor Carsten Lutz for the excellent supervision, in particular, for both giving me the liberty to work on my own projects and supporting me with many great ideas while collaborating. Apart from learning about computer science, I also learnt a great deal about how to do research.

Besides Carsten, I want to thank the other members of the working group “Theorie der künstlichen Intelligenz” at University Bremen: İnanç, Marlott, Maxim, Peter, Stefan, Thomas, Víctor, and Yazmin. It was (and is) a lot of fun working with you and I enjoyed the many discussions about diverse topics. Two people stand out: Víctor, with whom I shared office during the entire PhD, thanks for so many discussions, background knowledge, motivation, support, and fun during that time. Special thanks go to Stefan: not only you never gave up telling me about verification of infinite state systems, but also we had “etliche Biere” and countless conversations about science, work, and life.

I owe thanks to Diego Calvanese for agreeing to review the thesis and coming to Bremen for the defense. I would also like to thank the Deutsche Forschungsgemeinschaft for supporting the project “ProbDL” that I was employed on.

When I came to Bremen I was so lucky to move into the “WG”. Warm thanks go to Alja, Bretti, Jana, Laura, Ney, Timm, Tinnen, and Alex for these incredible years spent on the veranda, in the garden, kitchen, or living room, and for being like a family to me. Special thanks go to Jana and my Mum and sister for the continuous support, care, and always open ears.

Contents

1	Introduction	1
1.1	Logic and Uncertainty	2
1.2	Products of Modal Logics	9
1.3	Structure of the Thesis	11
1.4	Summary of Publications	13
2	Preliminaries	15
2.1	First-Order Logic	15
2.2	Description Logics	17
2.3	Probabilistic Description Logics	20
3	Monodic Fragments of Probabilistic First-order Logic	25
3.1	Probabilistic First-order Logic	28
3.1.1	Syntax and Semantics	28
3.1.2	Examples and Expressivity	31
3.1.3	Complexity and first Observations	33
3.2	Monodic ProbFO	41
3.2.1	Examples and Expressivity	42
3.2.2	Equality	44
3.3	The Quasi-Model Machinery	45
3.4	Recursive Enumerability and Axiomatization	53
3.5	Decidability and Complexity	59
3.5.1	Improvements	61
3.6	Connection to Probabilistic Description Logics	66
3.7	Conclusion and Outlook	68
4	Subjective Uncertainty in \mathcal{EL}	71
4.1	Syntax and Semantics of Prob \mathcal{EL}	74
4.2	Complexity of Probabilistic Concepts	77
4.2.1	Lower bounds	77
4.2.2	Subsumption Relative to Classical TBoxes	84
4.3	Complexity of Probabilistic Roles	93
4.3.1	Subsumption in full Prob \mathcal{EL}	93
4.3.2	Subsumption in Prob \mathcal{EL}_{01} is PSPACE-complete	94

4.4	Conclusion and Outlook	107
5	Ontology-Based Access to Probabilistic Data	109
5.1	Preliminaries	112
5.2	The Framework of Probabilistic OBDA	114
5.2.1	Computational Problems	117
5.2.2	Assertion-independent probabilistic ABoxes	119
5.3	The Dichotomy for First-Order Rewritable (φ, \mathcal{T})	121
5.4	The Dichotomy for DL-Lite TBoxes	125
5.4.1	Dichotomy for disjunctive sentences	127
5.4.2	Dichotomy for DL-Lite	135
5.5	Beyond First-order Rewritings: \mathcal{ELI} -TBoxes	140
5.5.1	TBox Normalization	143
5.5.2	Fixpoint operator, Boundedness, and FO-rewritability	145
5.5.3	Construction of $\widehat{\mathcal{A}}_d$ from Lemma 5.33	148
5.6	Monte Carlo Approximation	152
5.7	Conclusion and Future Directions	155
6	Computational Complexity of the Product Logic $\mathbf{K} \times \mathbf{K}$	159
6.1	Preliminaries	161
6.1.1	Many-dimensional modal logics	161
6.1.2	Decision problems	164
6.1.3	Bisimulation equivalence	167
6.2	\mathbf{K}^2 -SAT is hard for nonelementary time	168
6.2.1	Trees encoding numbers	168
6.2.2	Formulas enforcing the trees $\Upsilon_{\ell,n}(j)$	171
6.2.3	ℓ -NEXPTIME-hardness for each $\ell \geq 1$	178
6.3	Hardness results for $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$	184
6.4	Conclusions, open problems, further applications	190
7	Conclusion and Outlook	193
	List of Figures	197
	Bibliography	199

1 Introduction

Logic has been proved to be fundamental to the formalization and solution of many important problems in different branches of computer science. One of the most important and successful applications are *ontologies* which are used to model the terminology of an area of interest, like the medical domain, genetics, bioinformatics, and many more. Logic provides the theoretical foundation of many ontology languages such as first-order logic (FO) or description logics (DLs). Thus, ontologies have a well-defined semantics which enables automated reasoning supporting both the designer and the user of the ontology by making implicit knowledge explicit. As a consequence, ontologies play a fundamental role in the areas of knowledge representation and reasoning, Semantic Web, and databases.

While classical logics like FO and DLs are well-suited to represent and reason about static knowledge, they have shortcomings regarding their expressive power when it comes to modeling of and reasoning about uncertainty or dynamic aspects such as time or change. Thus, it is highly relevant to investigate principal ways of how to extend classical logic to capture these aspects. Arguably, the most prominent approach towards this direction is to adopt a *possible world semantics* known from modal logics. More specifically, we move from *one* formal representation of the real world – as it is standard in classical logics – to *many*. Intuitively, classical logic is used to reason inside one world, whereas another formalism which addresses the dynamic aspects is used to reason about the worlds seen as entities. As an example, uncertainty is often dealt with by equipping the worlds with probabilities, expressing how likely the world is considered to be.

Adding uncertainty or a dynamic aspect to classical logic can be viewed as adding an additional dimension of reasoning, which explains the first part of title of this thesis: **Reasoning in Many Dimensions**. The second part **Uncertainty and Products of Modal Logics** indicates that we concentrate on two particular ways of extending classical logics. First, we study in a principled way two frameworks that combine logic with probability theory for handling uncertainty. We motivate these frameworks in the realms of ontological modeling and ontology-based data access. Second, we study products of modal logics. In particular, we show the semantical connection to the first part of the thesis and solve some open problems from this field. Throughout the thesis, our main concern will be the study of the computational complexity of the relevant reasoning problems.

1.1 Logic and Uncertainty

Combining logic with uncertainty is an old and challenging problem in knowledge representation (KR) and artificial intelligence which has been subject of a lot of research. Depending on the addressed scenario, there is an abundance of choices about the nature of uncertainty, independence assumptions, the required logic, and so on, that has to be considered when developing such a combination. In order to illustrate the scope of this thesis and to provide some background, we will discuss some of these choices to be made. Note that we do not claim completeness here; the main goal is to give some general design aspects of combinations of logic with probabilities. For another (and more exhaustive) overview about classification of probabilistic logics, see [40].

Let us start with the *choice of the logic*. In this thesis, we will concentrate on two settings from KR involving ontologies, that is, logical description of the terminology of some domain of discourse. We will thus focus on first-order logic and description logics being well-known ontology languages; in particular, propositional logics are not considered in such a setting.

Next, let us have a look at the *nature of uncertainty*. It has been observed, for example in [67], that there are at least two forms of uncertainty that capture different ideas.

Subjective uncertainty This is uncertainty about particular objects, as in ‘The probability that Tweety flies is greater than 0.9’. Intuitively, statements of this kind capture the *degree of belief* that some object satisfies a property.

Statistical uncertainty This type uncertainty captures statistical information about the domain, such as ‘90 % of all birds fly’.

Semantically, statistical uncertainty, is typically modeled by a probability distribution over some domain; in the example about the birds, 90 % of the birds would additionally fly. Subjective uncertainty, on the other hand, is usually modeled in a possible worlds semantics, that is, using a probability distribution over a set of worlds. For a mathematician, this is a very natural point of view since possible worlds correspond to possible outcomes of random experiments. In combination with logic, however, this approach can be attributed to a seminal paper by Nilsson [110] who studied a setting involving (subjective) uncertainty of propositions. Intuitively, not only one world—a formal representation of the real world—is considered possible, but a collection of them. Moreover, the probability distribution over the worlds assigns probabilities to the worlds referring to the degree of belief that this is the actual ‘real’ world.

In the scope of this thesis, we will concentrate on the subjective view as motivated in the scenarios that we will present later. Having decided for a logic (a first-order language for ontological modeling) and subjective uncertainty (for representing degrees of belief), there are still several design choices and many approaches have been proposed for this setting. They can be roughly classified according to two criteria.

On the one hand, Halpern and Bacchus introduced *probabilistic first-order logic* (*ProbFO*) to combine first-order logic with means for capturing subjective uncertainty in a principled way [67, 13], and Lutz and Schröder recently studied description logic fragments thereof [101]. The distinctive feature is that these logics are proper extensions of their non-probabilistic versions, first-order logic and description logics. On the other hand, there is a whole range of formalisms in a different spirit. In a nutshell, they fix a finite domain and describe in a succinct way a fixed probability distribution over a set of possible worlds (which is determined by the domain). This corresponds to a Herbrand-style semantics which, technically, can be viewed as making the logic essentially propositional (although a first-order language is used). Examples for this are Markov logic [118] or some first-order generalizations of Bayesian networks [31], and many more; see [56, 40] for excellent overviews. Note that both fixing a domain and a set of worlds is inherently different from ProbFO where, intuitively, a formula describes *a class of possible distributions over possible worlds*.

The tool for the mentioned succinct representation of a probability distribution are *independence assumptions*. Independence of two statements or events S_1, S_2 in the domain intuitively refers to the fact that knowing about S_1 does not change the beliefs of S_2 . For example, a doctor's belief that a smoker will develop lung cancer is independent from whether the smoker is male or female. Independence assumptions have been successfully modeled in propositional settings, namely in Bayesian and Markov networks. Intuitively, these are graphical structures are used to encode all independence relations in the domain under consideration. For more details on probabilistic graphical models, we refer the interested reader to the textbooks [111, 91]. Here, we want to stress that being able to model the independence relations in the domain under consideration is typically a desired feature since independences are characteristic properties of the domain. As already mentioned, independence assumptions are inherent in the models of the second type, whereas this is not the case for ProbFO; however, they can be expressed.

In terms of efficient inference algorithms, research has been focused almost entirely on the second group of formalisms. Particularly, Abadi and Halpern have shown that ProbFO has prohibitive high complexity outside the arithmetic and analytic hierarchies [1]. With this in mind, Lutz and Schröder recently introduced a family of *probabilistic description logics* (*ProbDLs*) which relates to ProbFO in the same way as description logics relate to FO [101]. In particular, they exhibit much better complexity, mostly EXPTIME. These results motivate to take a fresh look at ProbFO and to further study ProbDLs and different applications thereof. Indeed, this is the main objective of the first part of this thesis. We next motivate and illustrate the research questions addressed in this thesis by sketching two realistic scenarios: uncertainty in ontologies and ontology-based access to probabilistic data.

Modeling Uncertainty in Ontologies

We consider the medical domain in which ontologies such as SNOMED CT and GALEN have been successfully used for modeling and classification. Many terms occurring in these ontologies can be precisely described in FO, take for example:

$$\begin{aligned}\forall x \text{ GastricMucosa}(x) &\leftrightarrow \text{Mucosa}(x) \wedge \exists y (\text{partOf}(x, y) \wedge \text{Stomach}(y)); \\ \forall x \text{ GastricUlcer}(x) &\leftrightarrow \text{Ulcer}(x) \wedge \exists y (\text{locatedAt}(x, y) \wedge \text{Stomach}(y)).\end{aligned}$$

Intuitively, the former defines the term *gastric mucosa*, to be the ‘mucosa that is the inner part of the stomach’ while the latter defines the term *gastric ulcer* as an ‘ulcer located at the stomach.’ Somewhat more complex, we can express that, if something is located at a subpart of some object, it is also located at that object:

$$\forall xyz \text{ locatedAt}(x, y) \wedge \text{partOf}(y, z) \rightarrow \text{locatedAt}(x, z)$$

For instance, an ulcer located at the gastric mucosa is also located at the stomach. In particular, from the above statements we can infer that indeed every ulcer located at the gastric mucosa is a gastric ulcer, that is,

$$\forall x (\text{Ulcer}(x) \wedge \exists y (\text{locatedAt}(x, y) \wedge \text{GastricMucosa}(y)) \rightarrow \text{GastricUlcer}(x).$$

The usefulness of logic lies in the fact that obviously the illustrated pattern applies not only to gastric ulcers. Much in the same way, we can infer that a cancer that is located at the left lung is a lung tumor, by specifying that the left lung is part of the lung.

The medical domain involves considerable uncertainty, both of statistical and subjective nature, and FO lacks a built-in means to capture it. For an instance of subjective uncertainty, notice that an oncologist is not always certain about the malignancy of a tumor; based on his experience, he rather has a certain degree of belief about some tumor to be benign, premalignant, or malignant. For an instance of statistical uncertainty, observe that as a matter of fact, not *all* lung cancer patients are smokers. However, there might be statistics that *75% of all patients suffering from lung cancer are smokers*.

As already advertised, we will concentrate on subjective uncertainty throughout this thesis. To further underpin the need, let us mention that in SNOMED CT, uncertain terms have been “modeled” by giving them names such as ‘probably malignant tumor’ or ‘natural death with probable cause suspected’. Obviously, these terms include subjective uncertainty which is not reflected in the semantics. Let us illustrate this with a small example. Assume we have stated that every *lung tumor* is a *tumor*:

$$\forall x \text{ LungTumor}(x) \rightarrow \text{Tumor}(x)$$

A desired consequence of this is that a probable malignant lung tumor is also a probably malignant tumor, that is, we would like to conclude

$$\forall x \text{ ProbablyMalignantLungTumor}(x) \rightarrow \text{ProbablyMalignantTumor}(x), \quad (*)$$

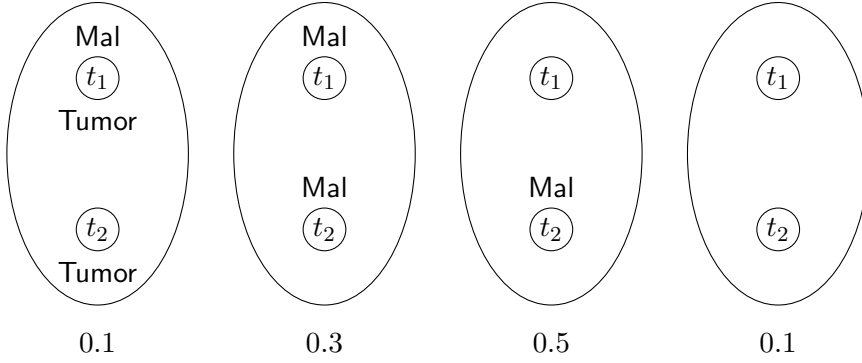


Figure 1.1: Example for the possible worlds semantics.

which is clearly not implied. A possible way to repair this is introducing a new concept name **ProbablyMalignant**. However, this is not a satisfying solution, since there is no built-in means for expressing different probabilities of being malignant or for comparing probabilities.

Halpern's and Bacchus' probabilistic first-order logic, ProbFO, provides explicit means to model degrees of beliefs. For instance, we can define the term *probably malignant lung tumor* as:

$$\forall x \text{ ProbablyMalignantLungTumor}(x) \leftrightarrow \text{LungTumor}(x) \wedge w(\text{Malignant}(x)) \geq 0.75,$$

expressing that a *probably malignant lung tumor* is defined as a *lung tumor* which we believe to be malignant with degree at least 0.75. Now, having an analogous definition for *probable malignant tumor*, we indeed get the desired consequence (*).

Let us explain the possible worlds semantics underlying ProbFO by means of an example. Assume two tumors t_1, t_2 for which malignancy is uncertain. This uncertainty is reflected by considering four possible worlds together with their degree of belief, see Figure 1.1. For each t_i , it is additionally indicated whether it is malignant, i.e., instance of **Mal** in the respective world. In the example, t_1 is malignant in the left two worlds, thus with probability $0.1 + 0.3 = 0.4$. In contrast, t_2 is an instance of **Mal** in the middle two worlds and hence with probability $0.3 + 0.5 = 0.8$. This means that t_2 is a *probably malignant tumor* and t_1 is not. Clearly, this is not the only distribution over possible worlds satisfying this condition; for example, one can assign probabilities of 0.4 and 0.4 to the two middle worlds. In this spirit, a ProbFO-ontology describes possible distributions over possible worlds, and neither the set of individuals nor the set of worlds is fixed.

ProbFO is a very general language with an immense expressive power and thus able to encompass many other probabilistic logics. In fact, it can be regarded as a 'baseline formalism' for other probabilistic models and logics, much in the same way as classic

first-order logic provides a baseline for other classic logics. However, the expressive power comes at a price: the standard reasoning problem of *satisfiability* has a prohibitive high complexity outside the arithmetical hierarchy, which is one reason why it has been mostly disregarded in practice. However, motivated by the recent positive results about the mentioned family of ProbDLs, it is highly interesting and relevant to revisit ProbFO with respect to computational complexity. In particular, we try to pinpoint the reason for ProbFO being so wildly undecidable, and try to identify other useful fragments. Hence, we are going to address the following research questions which are, as argued, important in the field of probabilistic logics.

Can we identify other “well-behaved” fragments of probabilistic first-order logics? What are maximal decidable fragments? How can we explain the good computational behavior of probabilistic description logics?

Given that we are seeking *maximal* decidable fragments of ProbFO, one cannot expect the answers to be of immediate practical relevance. Naturally, such fragments will exhibit high complexity. Thus, in a second step we move towards the other end of the expressivity (and thus complexity) scale and try to identify tractable fragments of ProbFO. In (non-probabilistic) description logics, tractability was achieved by studying the positive, existential fragment, \mathcal{EL} , of the basic description logic \mathcal{ALC} [9]. Although being of restricted expressivity, \mathcal{EL} is used as a basis for the aforementioned biomedical ontologies SNOMED CT and parts of GALEN. In particular, some of the above logical statements can actually be rewritten in \mathcal{EL} , for example:

$$\begin{aligned} \text{GastricMucosa} &\equiv \text{Mucosa} \sqcap \exists \text{partOf.Stomach} \\ \text{LungTumor} &\sqsubseteq \text{Tumor}. \end{aligned}$$

All this motivated the introduction of a probabilistic variant of \mathcal{EL} as member of Lutz’ and Schröder’s family of ProbDLs [101]. It has been shown, though, that reasoning in Prob \mathcal{EL} is as hard as in Prob \mathcal{ALC} , thus the syntactic restrictions do not result in better computational complexity. Given the need for practical probabilistic reasoning, it is interesting to take a closer look at fragments of Prob \mathcal{EL} . We investigate the following questions:

What is the computational complexity for reasoning in fragments of Prob \mathcal{EL} ? Are there any non-trivial fragments which offer polynomial time reasoning services and are still useful in practice?

Again, there are several choices of how to choose fragments of Prob \mathcal{EL} . In the scope of this thesis, we will concentrate on three possibilities. First, we will vary the application of probabilistic operators: is it only applied to concepts or also to roles? A second way is to constrain the possible values used as probabilities. Finally, we consider happens if the TBox language is restricted.

Ontology-based access to Probabilistic Data

A recently very popular application of ontologies is ontology-based data access, where the data is assumed to be incomplete and an ontology is used to retrieve facts that are only implicit in the data. To illustrate the idea, imagine a database D storing data about soccer players and their clubs, see Figure 1.2.

Player	playsfor	
Messi	Messi	FCBarcelona
Ronaldo	Ronaldo	RealMadrid

Figure 1.2: Some data about soccer players and their clubs.

An example query in this domain would be to ‘retrieve all players’, which is realized by the query

$$q(x) = \text{Player}(x).$$

As expected, the answers to $q(x)$ on database D are **Ronaldo** and **Messi**. Suppose now that there is an additional entry `playsfor(Casillas, RealMadrid)` and call the modified database D' . Obviously, the answers of $q(x)$ to D' remain the same. However, as typically people who play for some club are players, one would like to also get the answer **Casillas**. The framework of *ontology-based data access (OBDA)* resolves this problem by answering queries *relative to an ontology*. In the present example, we might have an ontology \mathcal{T} describing the soccer domain which contains the axiom that ‘everybody who plays for something is a player’, formulated in first-order logic as:

$$\forall x (\exists y \text{ playsfor}(x, y) \rightarrow \text{Player}(x)).$$

For answering the above query $q(x)$ relative to such an ontology, we switch to *certain answers*: we drop the closed-world assumption (which is standard in database settings), adopt the open-world assumption instead, and ask for all individuals that are players in *all models of \mathcal{T} and D'* . Clearly, this yields the additional answer **Casillas** since in every model of D' and \mathcal{T} , **Casillas** is a player. In a nutshell, OBDA uses an ontology of the application domain that serves as an interface for querying and allows to derive additional facts.

Imagine now we are aiming at a tool which manages data that is automatically extracted from the web. As such extracted data usually comes without explicit information about the involved relations (`Player, playsfor, ...`), it is typically incomplete. As demonstrated above, OBDA is a useful tool in this context. However, such extracted data is inherently *uncertain* which can be attributed to different reasons:

- *Ambiguity*. The tool processes metaphorical or ambiguous sentences, for example from a newspaper.

1 Introduction

- *Trust*. The tool might have different degrees of confidence in different web pages.
- *Currentness*. The tool processes a web page displaying data that is possibly outdated.

Note that in the mentioned items uncertainty is *subjective*: the tool assigns some degree of belief to the data it extracts. It has been argued that *probabilistic databases*, which intuitively assign a degree of belief to every tuple, are a suitable formalism to capture this uncertainty [125, 36]. For example, when the tool finds the information that Messi plays for Barcelona on Wikipedia it would associate a high degree of belief to the assertion `playsfor(Messi,FCBarcelona)`, since Wikipedia is typically correct. As another example, after parsing the ambiguous sentence ‘Messi is the soul of the Argentinian national soccer team’, the tool adds the assertion `Player(Messi)` with a medium degree of belief, since “soul” can also refer to the coach or the mascot. The result of this process is depicted Figure 1.3 where the p -column contains the degree of belief.

Player	p	playsfor	p
Messi	0.5	Messi FCBarcelona	0.9

Figure 1.3: Extracted data in a probabilistic database.

A probabilistic database such as the one in Figure 1.3 encodes a distribution over possible worlds as follows: each subset of the tuples is a world whose weight is given by considering the tuples as *independent events*. For example, the weight of the world $\{\text{Player(Messi)}\}$ consisting of a single tuple is $0.5 \times (1 - 0.9) = 0.05$. Thus, enabled by this independence of tuples, probabilistic databases are a succinct representation of a large distribution over possible worlds. Note that the justification for this independence assumption is that, in the setting of information extraction, tuples extracted from different (independent) sources can be regarded as independent.

Overall, we are facing the problem of *ontology-based access to probabilistic data* on which, so far, only little research has been conducted. In order to relate this setting to our baseline formalism ProbFO, let us point out that:

- uncertainty is (only!) in the data, that is, every tuple is equipped with a probability;
- the ontology is a classical (non-probabilistic) FO or DL ontology;
- we assume independence of all tuples in the (probabilistic) database;
- the set of worlds is fixed by the data: all possible subsets of the tuples;
- we assume open worlds; more precisely, we adopt the open world assumption in every single possible world.

The natural computational problem in such a setting is not computing certain answers (as in traditional OBDA), but *computing their probability*. Given the growing amounts of data in the web, applications like the sketched information extraction setting are typically data intensive, that is, the query and the ontology are small compared to the data. It has been argued that the right complexity measure is *data complexity* where query and ontology are fixed and the input consists only of the database, e.g., [127, 36, 28, 27]. While in traditional OBDA, ‘hardness’ of query answering is often characterized as (co)NP-hardness, we will use #P-hardness as the natural analog in probability computation problems.¹ A particular way to study complexity of query answering is the *non-uniform approach*, where each fixed ontology and/or query defines a single computational problem, and which has recently been proved central for both OBDA and pure probabilistic databases [102, 37]. The benefit of this approach is a better understanding of which conditions lead to tractability of queries. Most interesting in this context is the search for *dichotomies* which are theorems saying that each query answering problem (from a fixed class) is either in PTIME or NP (respectively, #P) hard. As an example, a dichotomy between PTIME and #P was recently obtained for unions of conjunctive queries and probabilistic databases by Dalvi, Schnaitter, and Suciu [37, 33]. In summary, this motivates to investigate the following research questions for OBDA to probabilistic data:

How can we define a framework of ontology-based access to probabilistic data as the combination of traditional OBDA and probabilistic databases? What is the computational complexity of query answering in the defined framework? How can we characterize tractable/hard queries? Can we prove dichotomy results for query answering?

1.2 Products of Modal Logics

An alternative approach of extending classical logics has been taken in the field of *many-dimensional modal logics* [52]. Motivated by the good computational behaviour of modal logics on the one hand and the need to talk about different modalities inside one application on the other hand, various combinations of different modal logics have been studied. A typical combination is modeling and reasoning about the evolution of knowledge over time. Note that there are modal logics for the individual domains—temporal logic and epistemic logic [43, 20]—but the question is how to combine them.

Several techniques for combining modal logics have been studied, the most fundamental ones being *fusions* and *products*. Intuitively, the fusion of two logics corresponds to a lightweight, *largely independent* combination of the logics. Due to that independence,

¹Strictly speaking, #P is a complexity class for counting problems, but counting and probability computation problems are closely related.

fusions often lead to computationally well-behaved logics. For example, the fusion of basic modal logic \mathbf{K} with itself is the bimodal logic \mathbf{K}_2 which is of the same complexity (for satisfiability), PSPACE [96]. Products, on the other hand, involve a rather strong interaction between the logics which makes products more complex than fusions. In this thesis, we will concentrate on products; we refer the reader interested in fusions to [52, Chapter 4].

The semantics of products of modal logics is given in terms of Kripke structures whose underlying frames are restricted to be the *direct product* of two frames. We denote with $\mathcal{L}_1 \times \mathcal{L}_2$ the product of the two modal logics $\mathcal{L}_1, \mathcal{L}_2$. The classes of frames that are considered are determined by the component logics. For instance, in $\mathbf{K} \times \mathbf{K}$, there are no restrictions on the frames while in $\mathbf{K4} \times \mathbf{K4}$, where $\mathbf{K4}$ is the variant of \mathbf{K} for reasoning over transitive frames, only products of two transitive frames are considered. The most relevant reasoning problem is *satisfiability checking* and it has been shown that the computational complexity in the product of two logics is often considerably higher than in the component logics. As an example, consider \mathbf{K} and its variant $\mathbf{K4}$, for both of which satisfiability is PSPACE-complete [96]. In contrast, only nonelementary upper bounds were known for $\mathbf{K} \times \mathbf{K}$ and $\mathbf{K4} \times \mathbf{K}$ [54, 105]. Even worse, satisfiability becomes undecidable in $\mathbf{K} \times \mathbf{K} \times \mathbf{K}$ [74] and $\mathbf{K4} \times \mathbf{K4}$ [55].

Although ProbFO and products of modal logics are seemingly unrelated, there is a deep semantical connection. Probabilistic first-order logic can be viewed as a *first-order modal logic*, that is, there is a modal operator—in this case the weight operator $w(\varphi)$ —that can speak about first-order formulas [68]. Note that further first-order modal logics like temporal or epistemic first-order logics have been studied [52, Part III]. These logics can express statements such as $\Diamond\varphi$ (‘at some point in the future φ holds’), or $K_i\varphi$ (‘some agent i knows φ ’). The basic observation, however, is that the semantics of these first-order modal logics is ‘product like’. This claim can be justified by considering modal description logics which are first-order modal logics whose first-order part is restricted to description logics. As a matter of fact, many complexity results about modal description logics require similar techniques as for products of modal logics; some can even be proved via reductions. As prominent example (for the former), let us mention that decidability of the modal description logic $\mathbf{K}_{\mathcal{ALC}}$ is proved by extending the technique for decidability of $\mathbf{K} \times \mathbf{K}$ [130].

The goal of this chapter is to study the precise complexity of the product logic $\mathbf{K} \times \mathbf{K}$ and some related logics, e.g., $\mathbf{K4} \times \mathbf{K}$. We already mentioned that decidability in nonelementary time was established and add here that the best lower bound previously known was NEXPTIME from [105]. In fact, determining the precise complexity of satisfiability in $\mathbf{K} \times \mathbf{K}$ was mentioned as important open problem in the standard textbook about many-dimensional modal logics [52]; it was conjectured to be hard for nonelementary time [105]. We confirm this conjecture by proving that:

Satisfiability in $\mathbf{K} \times \mathbf{K}$ is complete for nonelementary time.

1.3 Structure of the Thesis

Apart from the introduction, preliminaries, and the conclusion, this thesis can be divided into three main parts. The first part, Chapters 3 and 4, deals with well-behaved fragments of probabilistic first-order logic and addresses the questions formulated in the first scenario. In the second part, Chapter 5, we lay out the framework of ontology-based access to probabilistic data sketched above and study computational complexity in that framework. Finally, in Chapter 6, we give a nonelementary lower bound for the two-dimensional modal logic $\mathbf{K} \times \mathbf{K}$ and some variants thereof. In each of the chapters, we account for detailed related work and bibliographic references. More precisely, the thesis is structured as follows:

Chapter 2 We introduce basic notions and results for first-order logic, the relevant description logics, and probabilistic description logics. In particular, we cover syntax and semantics of these logics, and the complexity of the basic reasoning problems. The preliminary chapter is rather short, as the chapters differ considerably in the sense that they often require orthogonal notions and techniques.

Chapter 3 The chapter is devoted to identifying well-behaved fragments of probabilistic first-order logic (ProbFO). We start with reviewing the complexity and show that classical approaches for getting decidability of first-order logic such as the restriction to two variables do not lead to well-behaved fragments. We then investigate *monodic* fragments of first-order probabilistic logics, where probabilistic operators are restricted to formulas with at most one free variable. We introduce a suitable abstraction – so-called quasi-models – from the possible world semantics and exploit it in order to prove that this fragment is well-behaved in the following sense: we show (i) recursive enumerability, (ii) axiomatizability, and (iii) restrictions to decidable fragments of first-order logic lead to decidable fragments of monodic ProbFO. Point (iii) is established in a general way treating many fragments at once, provided that realizability – a slight generalization of satisfiability – is decidable (which is the case for all standard fragments such as the guarded fragment). We then take a closer look at the computational complexity for the decidable fragments and propose two improvements to the general approach. This leads to tight complexity results in some cases; most notably, we show 2EXPTIME-completeness for the monodic probabilistic guarded fragment. We conclude the chapter by clarifying the relation to recently introduced probabilistic description logics.

Chapter 4 Having identified in a sense maximal decidable fragments, we move to the other end of the expressivity and complexity scale and try to identify tractable fragments based on the well-known tractable description logic \mathcal{EL} . In the first part, we show that reasoning relative to general TBoxes becomes EXPTIME-hard as soon as non-trivial probabilistic operators are allowed. Consequently, we restrict the

general to classical TBoxes and give a polynomial time algorithm for \mathcal{EL} extended with *one* arbitrary probabilistic operator. We then leave the monodic framework and consider probabilistic operators applied to roles which are binary predicates in FO. There, we show PSPACE-completeness for subsumption relative to Prob \mathcal{EL}_{01} -TBoxes, that is, the extension with the ‘probabilistic’ operators $P_{>0}$ and $P_{=1}$. We show maximality of the fragment in the sense that adding *any* probabilistic operator leads to 2EXPTIME-hardness while not even decidability is known.

Chapter 5 In this chapter, we motivate and introduce the framework of ontology-based access to probabilistic data. We define it as a natural probabilistic extension of classical ontology-based data access (OBDA). In this framework, we study the complexity of computing the probability of certain answers. In doing so, we pursue the non-uniform approach: each pair (q, \mathcal{T}) of query q and an ontology \mathcal{T} defines a reasoning problem $\text{pOBDA}(q, \mathcal{T})$ and we are interested in whether $\text{pOBDA}(q, \mathcal{T})$ is tractable or not, that is, $\#P$ -hard. We first have to restrict the input to so-called assertion-independent pABoxes, because otherwise each of these problems is intractable, giving rise to problems $\text{ipOBDA}(q, \mathcal{T})$. Based on a recent dichotomy result for answering unions of conjunctive queries on probabilistic databases and the concept of FO-rewritability known from classical OBDA, we provide several dichotomy theorems, such as: If the query q is FO-rewritable relative to the TBox \mathcal{T} , then $\text{ipOBDA}(q, \mathcal{T})$ is either in PTIME or $\#P$ -hard. We also show dichotomies for the TBox languages DL-Lite and \mathcal{ELI} and conjunctive queries. In the case of DL-Lite-TBoxes, we try to get a better understanding *which* pairs (q, \mathcal{T}) are tractable and are able to give a concrete classification. For \mathcal{ELI} -TBoxes, our main result is that FO-rewritability turns out to be a necessary condition for a query q being in PTIME, or viewed differently: proving PTIME of $\text{ipOBDA}(q, \mathcal{T})$ can always be done via FO-rewritings. We finally study Monte Carlo approximations to the problem $\text{pOBDA}(q, \mathcal{T})$ in the form of FPRASes and show that FO-rewritability often implies the existence of an FPRAS. Conversely, we show that non-FO-rewritability often implies non-approximability.

Chapter 6 We study satisfiability in several two-dimensional modal logics. In particular, we establish a nonelementary lower bound for the logic $\mathbf{K} \times \mathbf{K}$ and some variants thereof, thereby improving the previously known NEXPTIME-lower bound. Most of the chapter is devoted to the technique for proving the lower bound which is of independent interest. In particular, we first define a family of trees parametrized by non-negative integers ℓ, n such that a tree associated to ℓ, n has depth ℓ and is nonelementary branching in the depth ℓ (the precise influence of n will become clear later). Then, we provide a family of $\mathbf{K} \times \mathbf{K}$ -formulas that are satisfiable only in products of the mentioned trees. We use these formulas to encode arbitrarily big elementary counters, which enables us to do a reduction

from ℓ -fold exponential tiling problems. Consequently, we obtain a nonelementary lower bound for satisfiability in $\mathbf{K} \times \mathbf{K}$. Finally, we apply well-known reductions from classical, that is, one-dimensional modal logic to extend this lower bound to satisfiability in $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$.

Chapter 7 We conclude the thesis and sketch future research.

1.4 Summary of Publications

Most of the technical content of the thesis has appeared in journal, conference, or workshop proceedings. In detail:

Chapter 3

- [86] *Jean Christoph Jung, Carsten Lutz, Sergey Goncharov, Lutz Schröder*. Monodic Fragments of Probabilistic First-order Logic. In *Proceedings of the 41st International Conference on Automata, Languages, and Computation (ICALP 2014)*, 2014.

Chapter 4

- [65] *Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, Lutz Schröder*. A Closer Look at the Probabilistic Description Logic Prob-EL. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI 2011)*, 2011.
- [66] *Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, Lutz Schröder*. The Complexity of Probabilistic EL. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745 of CEUR-WS, 2011.

Chapter 5

- [84] *Jean Christoph Jung, Carsten Lutz*. Ontology-Based Access to Probabilistic Data with OWL-QL. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*. Springer, 2012.
- [85] *Jean Christoph Jung, Carsten Lutz*. Ontology-Based Access to Probabilistic Data. In *Proceedings of the 26th International Workshop on Description Logics (DL 2013)*, volume 1014 of CEUR-WS, 2013.

Chapter 6

- [57] *Stefan Göller, Jean Christoph Jung, Markus Lohrey.* The complexity of decomposing modal and first-order theories. In *Proceedings of the 27th ACM/IEEE Symposium on Logic in Computer Science (LICS 2012)*, ACM/IEEE, 2012.
- [58] *Stefan Göller, Jean Christoph Jung, Markus Lohrey.* The complexity of decomposing modal and first-order theories. In *ACM Transactions on Computational Logic*, to appear.

2 Preliminaries

In this chapter, we briefly introduce fundamental notions that are used throughout this thesis. Additionally, each chapter has its own preliminary section for notations that are exclusively used in that chapter. Here, we start with first-order logic and recall syntax, semantics, and some decidable fragments. Next, we briefly introduce description logics, focusing on \mathcal{ALCI} and some fragments, and on the relevant reasoning problems. Moreover, we introduce the mentioned probabilistic description logic $\text{Prob}\mathcal{ALC}$ and recall some complexity results obtained for them so far [101].

2.1 First-Order Logic

First-order logic (FO) is the most fundamental logic in computer science. The syntax of first-order logic is based on a *signature* containing predicate and constant symbols, where each predicate symbol comes with an arity. First-order formulas over a signature Σ are built according to the following syntax rule:

$$\varphi, \psi ::= R(t_1, \dots, t_k) \mid \neg\varphi \mid \varphi \wedge \psi \mid \exists x \varphi(x)$$

where $R \in \Sigma$ is a k -ary predicate symbol and each t_i is either a constant symbol from Σ or a variable (taken from a countably infinite supply of variable symbols). Note that we do not allow for function symbols except for constants. Sometimes we add *equality*, that is, atoms of the form $t = t'$ with object terms t, t' , and denote the corresponding extension of FO with $\text{FO}^=$. The semantics of FO is given in terms of relational structures $\mathfrak{A} = (A, \pi)$, where:

- A is the domain of \mathfrak{A} and
- π is the interpretation function assigning to each k -ary predicate symbol R a subset $\pi(R) \subseteq A^k$ and to each constant symbol c a domain element $\pi(c) \in A$.

A *valuation for \mathfrak{A}* is a function ν from the set of variables to the domain. The truth relation \models is now defined by induction on the structure of formulas:

$$\begin{aligned} (\mathfrak{A}, \nu) \models R(t_1, \dots, t_k) & \quad \text{if} \quad (a_1, \dots, a_k) \in \pi(R), \text{ where} \\ & \quad a_i \text{ is } \nu(t_i) \text{ if } t_i \text{ is a variable and } \pi(t_i) \text{ otherwise;} \\ (\mathfrak{A}, \nu) \models \neg\varphi & \quad \text{if} \quad \text{not } (\mathfrak{A}, \nu) \models \varphi; \end{aligned}$$

$(\mathfrak{A}, \nu) \models t_1 = t_2$	if	$a_1 = a_2$, where the a_i are defined as above;
$(\mathfrak{A}, \nu) \models \varphi \wedge \psi$	if	$(\mathfrak{A}, \nu) \models \varphi$ and $(\mathfrak{A}, \nu) \models \psi$;
$(\mathfrak{A}, \nu) \models \exists x \varphi(x)$	if	there is $a \in A$ with $(\mathfrak{A}, \nu[x/a]) \models \varphi(x)$.

We indicate with $\varphi(\vec{x})$ that φ might have free variables from \vec{x} and call formulas without free variables *sentences*. We say that a formula $\varphi(\vec{x})$ is *satisfiable* if there is a structure \mathfrak{A} and a valuation ν such that $(\mathfrak{A}, \nu) \models \varphi(\vec{x})$. For sentences φ , we drop the valuation and just write $\mathfrak{A} \models \varphi$. A sentence φ is *valid* if $\neg\varphi$ is not satisfiable. The associated reasoning problems are defined as follows.

SATISFIABILITY

INPUT: FO formula φ

OUTPUT: Is φ satisfiable?

VALIDITY

INPUT: FO formula φ

OUTPUT: Is φ valid?

It is well-known that both the satisfiability and the validity problem for FO are undecidable, and validity is recursively enumerable.

Decidable Fragments

Since satisfiability in FO is undecidable, researchers investigate (preferably expressive) fragments with a decidable satisfiability problem. Relevant for this thesis are:

- the *two-variable fragment* [62], where at most two variables are allowed;
- the *monadic fragment*, where only unary predicate symbols are allowed;
- the *guarded fragment (GF)* [60], which is defined as the minimal set satisfying the following:
 - (1) every atomic formula $R(t_1, \dots, t_k)$ belongs to GF;
 - (2) GF is closed under the connectives \wedge, \neg ;
 - (3) If \vec{x}, \vec{y} are tuples of variables, $\alpha(\vec{x}, \vec{y})$ is atomic (including equality $x = x$) and $\psi(\vec{x}, \vec{y})$ is a formula in GF with at most the free variables of the atom α , then the formula $\exists \vec{y} \alpha(\vec{x}, \vec{y}) \wedge \psi(\vec{x}, \vec{y})$ is guarded;
- the *guarded negation fragment (GNFO)* [15], which is defined by the following syntax rule, restricting the use of negation:

$$\varphi, \psi ::= R(t_1, \dots, t_k) \mid x = y \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \exists x \varphi(x) \mid \alpha(\vec{x}, \vec{y}) \wedge \neg\varphi(\vec{y})$$

where α is an atom $R(t_1, \dots, t_k)$ or $x = x$ containing all free variables of φ ;

- description logics, which are introduced in more detail below.

It is well-known that satisfiability in all these logics is decidable. More precisely, it is NEXPTIME-complete for the monadic and the two-variable fragment, and 2EXPTIME-complete for GF and GNFO. If the number of variables or the maximal arity of the predicate symbols is bounded, then satisfiability in GF becomes EXPTIME-complete.

2.2 Description Logics

Description logics (DLs) are a family of languages for knowledge representation and reasoning. They are subsets of first-order logic intended for modeling and reasoning under terminological and assertional knowledge and offer—in contrast to FO—decidable reasoning services. We only introduce the basic notions and mention relevant results; for more detailed information, consult [12].

Syntax and Semantics

We use standard notation for the syntax and semantics of description logics. Let N_C , N_R , and N_I denote countably infinite sets of concept names, role names, and individual names, respectively. We introduce \mathcal{ALCI} as our basic description logic. \mathcal{ALCI} -concepts are formed according to the following syntax rule:

$$C, D ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \exists R.C,$$

where A ranges over N_C and R is either a role name $r \in N_R$ or its *inverse* r^- . We use the abbreviations $\forall R.C$ for $\neg \exists R. \neg C$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, and \perp for $\neg \top$. The set of \mathcal{ALC} -, \mathcal{ELI} -, and \mathcal{EL} -concepts is defined by disallowing R to be an inverse role, dropping negation, and both, respectively, from the syntax rule for \mathcal{ALCI} -concepts.

Description logic knowledge bases are typically separated in *background* or *terminological knowledge* and *assertional knowledge*. The former is represented in the TBox while the latter, the data, is represented in an ABox. For $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALC}, \mathcal{ELI}, \mathcal{EL}\}$, a *general \mathcal{L} -TBox* is a set \mathcal{T} of *concept inclusions* $C \sqsubseteq D$ with C, D \mathcal{L} -concepts. A *classical \mathcal{L} -TBox* is a set \mathcal{T} of *concept definitions* $A \equiv C$ such that each concept name $A \in N_C$ occurs at most once in the left-hand side of a concept definition in \mathcal{T} and C is an \mathcal{L} -concept. We will drop the reference to the TBox language when no confusion is possible. Note that a classical TBox is a special case of a general TBox since we can replace $A \equiv C$ by the two concept inclusions $A \sqsubseteq C$, $C \sqsubseteq A$. An *ABox* is a set of *assertions* of the form $C(a)$ or $r(a, b)$ with C a concept description, $r \in N_R$, and $a, b \in N_I$. We denote the set of all individuals appearing in some ABox \mathcal{A} with $\text{Ind}(\mathcal{A})$. A DL knowledge base is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

The semantics of DLs is given through interpretations. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where

- $\Delta^{\mathcal{I}}$ is a non-empty set of individuals, the *domain*, and
- $\cdot^{\mathcal{I}}$ is an *interpretation function* mapping each $a \in \mathbf{N}_I$ to some domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $A \in \mathbf{N}_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of the domain and each role name $r \in \mathbf{N}_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ over the domain.

Throughout the thesis we make the *unique name assumption (UNA)*, that is, we assume that different individuals are interpreted by different domain elements. The interpretation function is extended to complex \mathcal{ALCI} concepts as follows:

$$\begin{aligned}\top^{\mathcal{I}} &= \Delta^{\mathcal{I}}; \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}; \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}; \\ (r^-)^{\mathcal{I}} &= \{(y, x) \mid (x, y) \in r^{\mathcal{I}}\}; \\ \exists R.C &= \{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d, e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}.\end{aligned}$$

An interpretation \mathcal{I} *satisfies* or *is a model of*

- a concept C if $C^{\mathcal{I}} \neq \emptyset$;
- a concept inclusion $C \sqsubseteq D$, written $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;
- a concept definition $A \equiv C$, written $\mathcal{I} \models A \equiv C$, if $A^{\mathcal{I}} = C^{\mathcal{I}}$;
- a general TBox \mathcal{T} , written $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I} \models C \sqsubseteq D$ for all $C \sqsubseteq D \in \mathcal{T}$;
- an ABox \mathcal{A} , written $\mathcal{I} \models \mathcal{A}$, if for each assertion $A(a) \in \mathcal{A}$, we have $a^{\mathcal{I}} \in A^{\mathcal{I}}$ and for each assertion $r(a, b) \in \mathcal{A}$, we have $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

Reasoning Problems and Complexity

Traditional reasoning problems for DLs are *concept satisfiability*, *knowledge base consistency*, and *subsumption*. We say that a concept C is *satisfiable relative to a TBox \mathcal{T}* if there is a common model of C and \mathcal{T} . A concept C is *subsumed by D relative to a TBox \mathcal{T}* , written $\mathcal{T} \models C \sqsubseteq D$, when for all models \mathcal{I} of \mathcal{T} we have $\mathcal{I} \models C \sqsubseteq D$. A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *consistent* if there is a common model of \mathcal{T} and \mathcal{A} . Thus, the mentioned reasoning problems are defined as follows:

CONCEPT SATISFIABILITY

INPUT: concept C , TBox \mathcal{T}

OUTPUT: Is C satisfiable relative to \mathcal{T} ?

KNOWLEDGE BASE CONSISTENCY

INPUT: Knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$

OUTPUT: Is \mathcal{K} consistent?

SUBSUMPTION

INPUT: concepts C, D and TBox \mathcal{T}

OUTPUT: Is C subsumed by D relative to \mathcal{T} ?

These problems will be instantiated to different DL dialects. It is well-known that for \mathcal{ALCI} and \mathcal{ALC} , KB consistency is the most general problem in the sense that the other problems can be reduced to it. In particular, C is satisfiable relative to \mathcal{T} iff the knowledge base $(\mathcal{T}, \{C(a)\})$ is consistent, and $\mathcal{T} \models C \sqsubseteq D$ iff $(\mathcal{T}, \{(C \sqcap \neg D)(a)\})$ is inconsistent. It is known that all problems are EXPTIME-complete in the presence of general and classical \mathcal{ALCI} - and \mathcal{ALC} -TBoxes [12].

In \mathcal{EL} and \mathcal{ELI} , every concept and every knowledge base is satisfiable as the logics lack negation \neg and bottom \perp . For this reason, subsumption becomes the standard reasoning problem. In fact, it has been shown that:

- checking subsumption relative to general \mathcal{ELI} -TBoxes is EXPTIME-complete [11];
- checking subsumption relative to general \mathcal{EL} -TBoxes (and various extensions) can be done in polynomial time [9].

DL-Lite

The DL-Lite family of description logics has been introduced to reason over database constraints imposed by conceptual data models such as ER and UML diagrams and for the purpose of ontology-based data access [28]. Members of the DL-Lite-family are arguably less expressive than most of the DLs mentioned above. However, DL-Lite is tailored to allow for *query rewriting*, a technique for ontology-based data access introduced in Chapter 5. Moreover, DL-Lite forms the logical underpinning of the OWL language for accessing databases, OWL2 QL.¹ A DL-Litebasic concept B is of the form

$$B ::= \top \mid \perp \mid A \mid \exists R$$

where A is a concept name and R is a role. Note that there is no nesting of concept constructors in DL-Lite. A DL-Lite-TBox is a finite set of *concept inclusions* $B \sqsubseteq B'$ and $B \sqcap B' \sqsubseteq \perp$ where B and B' are basic DL-Lite-concepts. This basic version is usually called $DL\text{-}Lite_{\text{core}}$. Sometimes, more expressive concept inclusions are used, such as:

- concept inclusions of the form $B_1 \sqcap \dots \sqcap B_n \sqsubseteq B$ with B_i, B basic concepts;
- *role inclusions* $R \sqsubseteq S$.

¹http://www.w3.org/TR/owl-profiles/#OWL_2_QL.

We refer with $DL-Lite_{\text{horn}}$ to the former extension of DL-Lite and add the superscript \mathcal{R} for the latter as, for example, in $DL-Lite_{\text{horn}}^{\mathcal{R}}$.

Semantics is given to DL-Lite-concepts and TBoxes in the same way as for the basic DLs by noting that the concept $\exists R$ is an abbreviation for $\exists R.\top$ and an interpretation \mathcal{I} satisfies a role inclusion $R \sqsubseteq S$ if $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.

2.3 Probabilistic Description Logics

We next introduce the family of probabilistic description logics (ProbDLs) that was recently developed by Lutz and Schröder [101] with the aim to enrich classical DLs with means for expressing subjective uncertainty. Intuitively, ProbDLs correspond to ProbFO [67, 13] just as DLs correspond to first-order logic. Let us start with the syntax and semantics. *ProbALC*-concepts are formed by extending the syntax rule for *ALC*; in particular, we allow to apply *probabilistic operators* to both concepts and roles.

$$C, D ::= A \mid \exists r.C \mid \neg C \mid C \sqcap D \mid P_{\sim p}C \mid \exists P_{\sim p}r.C,$$

where $p \in [0, 1]$ and $\sim \in \{<, \leq, =, \geq, >\}$. We call $P_{\sim p}C$ a *probabilistic concept* and $P_{\sim p}r$ a *probabilistic role*. The notion of general TBox is extended in a straightforward way to ProbALC, that is, a general ProbALC-TBox is a collection of concept inclusions $C \sqsubseteq D$ with C, D ProbALC-concepts.

Probabilistic ABoxes store the knowledge we are having about the instances. They are expressions formed according to the rule

$$\mathcal{A} ::= C(a) \mid r(a, b) \mid \neg \mathcal{A} \mid \mathcal{A} \wedge \mathcal{A}' \mid P_{\sim p}\mathcal{A}$$

where C, r, \sim , and p are as above, $a, b \in \mathbf{N}_I$, and $\mathcal{A}, \mathcal{A}'$ range over probabilistic ABoxes. A *knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with \mathcal{T} a TBox and \mathcal{A} an ABox. Note that, in contrast to non-probabilistic DLs, we allow probabilistic operators to be applied to ABoxes. To take full advantage of this operator and speak about the probability of more than a single fact such as in $P_{\geq 0.5}A(a) \wedge P_{\geq 0.1}(r(a, b) \wedge B(b))$, we also include Boolean connectives as ABox operators.

To provide a semantics for ProbALC, we are using a *possible worlds semantics*, that is, we generalize interpretations to probabilistic interpretations. Formally, a *probabilistic interpretation* takes the form $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$, where $\Delta^{\mathcal{I}}$ is the (non-empty) domain, W a non-empty set of *possible worlds*, μ a discrete probability distribution on W , and for each $w \in W$, \mathcal{I}_w is a classical DL interpretation with domain $\Delta^{\mathcal{I}}$. We assume *rigid constants*, that is, $a^{\mathcal{I}_w} = a^{\mathcal{I}_{w'}}$ for all $a \in \mathbf{N}_I$ and $w, w' \in W$. Since $a^{\mathcal{I}_w}$ does not depend on w , we write only $a^{\mathcal{I}}$. We usually write $C^{\mathcal{I}, w}$ for $C^{\mathcal{I}_w}$, and likewise for $r^{\mathcal{I}, w}$. For concept names A and role names d , we define the probability

- $p_d^{\mathcal{I}}(A)$ that $d \in \Delta^{\mathcal{I}}$ is an A as $\mu(\{w \in W \mid d \in A^{\mathcal{I}, w}\})$;

- $p_{d,e}^{\mathcal{I}}(r)$ that $d, e \in \Delta^{\mathcal{I}}$ are related by r as $\mu(\{w \in W \mid (d, e) \in r^{\mathcal{I},w}\})$.

Next, we extend $p_d^{\mathcal{I}}(A)$ to compound concepts C and define the extension $C^{\mathcal{I},w}$ of compound concepts by mutual induction on C . The definition of $p_d^{\mathcal{I}}(C)$ is exactly as in the base case, with A replaced by C . The extension of compound concepts is defined as follows:

$$\begin{aligned} (\neg C)^{\mathcal{I},w} &= \{d \in \Delta^{\mathcal{I}} \mid d \notin C^{\mathcal{I},w}\} \\ (C \sqcap D)^{\mathcal{I},w} &= \{d \in \Delta^{\mathcal{I}} \mid d \in C^{\mathcal{I},w} \text{ and } d \in D^{\mathcal{I},w}\} \\ (\exists r.C)^{\mathcal{I},w} &= \{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I},w} : (d, e) \in r^{\mathcal{I},w}\} \\ (P_{\sim p}C)^{\mathcal{I},w} &= \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) \sim p\} \\ (\exists P_{\sim p}r.C)^{\mathcal{I},w} &= \{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I},w} : p_{d,e}^{\mathcal{I}}(r) \sim p\} \end{aligned}$$

A probabilistic interpretation \mathcal{I} *satisfies* a concept inclusion $C \sqsubseteq D$ (written $\mathcal{I} \models C \sqsubseteq D$) if $C^{\mathcal{I},w} \subseteq D^{\mathcal{I},w}$ for all worlds w . It is a *model* of a TBox \mathcal{T} if it satisfies all concept inclusions in \mathcal{T} .

To give a semantics to probabilistic ABoxes \mathcal{A} , we again use mutual induction, defining the probability $p^{\mathcal{I}}(\mathcal{A})$ that \mathcal{A} is true as

$$p^{\mathcal{I}}(\mathcal{A}) = \mu(\{w \in W \mid \mathcal{I}, w \models \mathcal{A}\})$$

and defining when a world w of \mathcal{I} *satisfies* \mathcal{A} (written $\mathcal{I}, w \models \mathcal{A}$) as follows:

$$\begin{aligned} \mathcal{I}, w \models C(a) &\quad \text{iff} \quad a^{\mathcal{I}} \in C^{\mathcal{I},w} \\ \mathcal{I}, w \models r(a, b) &\quad \text{iff} \quad (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I},w} \\ \mathcal{I}, w \models \neg \mathcal{A} &\quad \text{iff} \quad \mathcal{I}, w \not\models \mathcal{A} \\ \mathcal{I}, w \models \mathcal{A} \wedge \mathcal{A}' &\quad \text{iff} \quad \mathcal{I}, w \models \mathcal{A} \text{ and } \mathcal{I}, w \models \mathcal{A}' \\ \mathcal{I}, w \models P_{\sim p}(\mathcal{A}) &\quad \text{iff} \quad p^{\mathcal{I}}(\mathcal{A}) \sim p \end{aligned}$$

We say that \mathcal{I} is a *model* of \mathcal{A} if $\mathcal{I}, w \models \mathcal{A}$ for some world w . It is a *model* of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of both \mathcal{T} and \mathcal{A} . We say that a knowledge base \mathcal{K} is *consistent* if it has a model. This gives rise to the corresponding problem of deciding *KB consistency*, defined as for classical \mathcal{ALC} .

Example 2.1. Consider the probabilistic interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ depicted in Figure 2.1. In particular, we have

- $\Delta^{\mathcal{I}} = \{a, b, c\}$;
- $W = \{w_0, w_1, w_2\}$ with $\mu(w_0) + \mu(w_1) + \mu(w_2) = 1$;
- $A^{\mathcal{I},w_0} = \{a\}$, $A^{\mathcal{I},w_1} = A^{\mathcal{I},w_2} = \emptyset$, and similarly for B , C , and r .

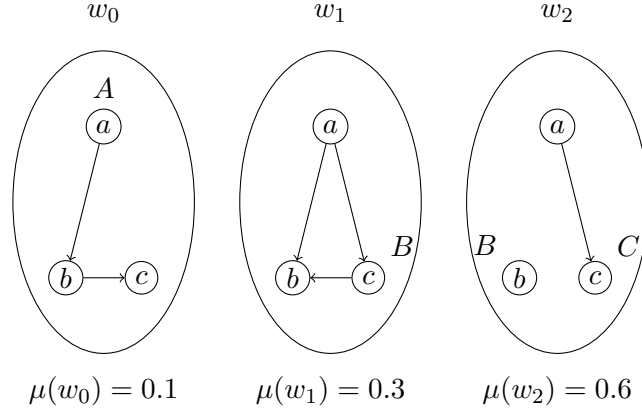


Figure 2.1: Example for a probabilistic interpretation.

Considering probabilistic concepts, we have, for example, $p_a^{\mathcal{I}}(A) = \mu(w_0) = 0.1$ and hence $a \in (P_{\leq p}A)^{\mathcal{I},w}$ for all $w \in W$ and $p \geq 0.1$. For probabilistic roles, we have for instance

$$p_{a,b}^{\mathcal{I}}(r) = 0.4 \quad \text{and} \quad p_{a,c}^{\mathcal{I}}(r) = 0.9$$

and thus $a \in (\exists P_{>0.1}r.B)^{\mathcal{I},w_1}$ and $a \in (\exists P_{>0.1}r.B)^{\mathcal{I},w_2}$. Note, however, that $a \notin (\exists P_{>0.1}r.B)^{\mathcal{I},w_0}$ as $B^{\mathcal{I},w_0} = \emptyset$.

Examples for Modeling with ProbALC

The probabilistic operator enables us to describe concepts involving uncertainty (in the TBox) and uncertainty of instance data (in the ABox). All our examples are taken from the medical domain, since uncertainty is pervasive there. For example, the medical ontology SNOMED CT involves concept *names* indicating uncertainty such as ‘animal bite by potentially rabid animal’ and ‘disease of possible viral origin’. We can model the former as

$$\text{Bite} \sqcap \exists \text{by}.(\text{Animal} \sqcap P_{>0.5} \exists \text{has.Rabies}),$$

where the probabilistic constructor $P_{>0.5}$ is applied to the concept $\exists \text{has.Rabies}$. The latter concept can be modeled using a probabilistic role as

$$\text{Disease} \sqcap \exists P_{>0} \text{origin.Viral}.$$

Concept inclusions involving probabilistic concepts can, for instance, be used to model that some diseases, even if not diagnosed with certainty, should be treated since they are very severe. An example for this situation [101] is:

$$P_{\geq 0.8}(\exists \text{hasDisease.LymeDisease}) \sqsubseteq \exists \text{recommendedTreatment.Antibiotics}.$$

ABoxes are used to describe instance data, both non-probabilistic such as

Patient(john), Fever(f), and hasSymptom(john, f),

and probabilistic such as

$P_{\leq 0.01}(\exists \text{hasCause.Malaria}(f))$ and $P_{\geq 0.9}(\exists \text{hasCause.Flu}(f))$.

The latter two assertions express that the fever is probably not caused by malaria, but probably caused by a flu. This might be the typical diagnosis, when the patient John has not been out of Europe. In contrast, one might assert

$P_{\geq 0.7}(\exists \text{hasCause.}(\text{Malaria} \sqcup \text{JapaneseEncephalitis}))(f)$

when John was traveling in the Malaysian jungle, which intuitively expresses that the probability that John's fever is caused by malaria or Japanese encephalitis is comparably high.

Complexity of Reasoning in Prob \mathcal{ALC}

The complexity of deciding KB consistency for Prob \mathcal{ALC} is open. There are, however, results for some useful fragments. We state here only the relevant results; for more details and further information, consult [101]. Let Prob \mathcal{ALC}_c be the fragment that dispenses with probabilistic roles, that is, the constructor $\exists P_{\sim p} r.C$ is dropped. It was shown that reasoning in this fragment is no more difficult than in the base logic \mathcal{ALC} .

Theorem 2.2. *Deciding consistency of Prob \mathcal{ALC}_c knowledge bases is EXPTIME-complete.*

Let us denote with Prob \mathcal{ALC}_{01} the fragment of Prob \mathcal{ALC} that allows only probabilistic operators $P_{>0}$ and $P_{=1}$, applied to both concepts and roles.

Theorem 2.3. *Deciding consistency of Prob \mathcal{ALC}_{01} knowledge bases is 2EXPTIME-complete.*

Thus, reasoning including probabilistic roles is more complex, even when only qualitative probabilistic operators are allowed. In fact, reasoning quickly becomes undecidable, for example when means for expressing linear equalities or independence constraints are added. In contrast, such constructors can be added to Prob \mathcal{ALC}_c without changing the complexity [101].

3 Monodic Fragments of Probabilistic First-order Logic

In the introduction of the thesis, we have illustrated the difficulties in modeling and ontological reasoning in domains involving uncertainty. We thus motivated combinations of logic with probabilities. In the 1990s, Halpern and Bacchus introduced a natural and fundamental such combination by enriching classical first-order logic (FO) with a probabilistic component [67, 13, 14]. The introduced probabilistic first-order logics (ProbFOs) come in essentially two versions reflecting the two main types of uncertainty and one version for their combination:

- *type-1 ProbFO* is used to reason about *statistical* probabilities. This is modeled in the semantics by a probability distribution over the domain of a classical FO structure.
- *type-2 ProbFO* is used to reason about *subjective* probabilities or degrees of belief. Semantically, this is reflected in a *possible world semantics*, that is, a probability distribution over a collection of possible worlds.
- *type-3 ProbFO* is the combination of the above and features both a distribution over the domain and a distribution over possible worlds.

Since we consider only subjective probabilities throughout the thesis, we will in this chapter concentrate on type-2 and will from now on generally use ‘ProbFO’ to refer to ‘type-2 ProbFO’. Type-1 and type-3 ProbFO will be mentioned again at the end of the chapter.

Although reasoning in ProbFO is of course undecidable—it contains full first-order logic—it is still useful as a general and uniform ‘baseline formalism’ that encompasses many other probabilistic logics, much in the same way that FO provides a baseline formalism for many other logics used in computer science. However, traditional ProbFO is not only undecidable, but computationally much less well-behaved than classical FO. Its disastrous computational behaviour was analyzed by Abadi and Halpern, who showed that validity is Π_1^2 -complete [1], thus outside the arithmetic and analytic hierarchies and, in particular, far from being recursively enumerable. This result holds up even when only unary predicates are admitted. A notable exception to the prohibitive high complexity in this framework is a family of *probabilistic description logics* recently introduced by Lutz and Schröder [101, 65] which exhibit lower complexity, mostly EXPTIME.

Motivated by this, our aim in this chapter is to revisit the computational complexity of ProbFO and to analyze how and how far the problematic computational properties of ProbFO can be improved. Our specific goals are to:

- identify a fragment of ProbFO with a recursively enumerable validity problem in order to enable theorem proving;
- identify maximal decidable fragments of ProbFO; and
- seek an explanation of the good computational properties of the mentioned family of probabilistic description logics.

Related Work

The combination of logic and probability theory involves a large number of choices and trade-offs, which has resulted in a broad spectrum of formalisms that vary greatly in spirit, semantics, and expressive power. Notably, this includes the choice of the ‘right character’ of uncertainty (subjective, statistical, ...). As argued before, this thesis is mostly confined to the subjective view, so we will mention here only approaches for modeling degrees of belief. The vast majority of such proposals is based on (some form of) the possible world semantics. Many of the early works from the 1980s are propositional such as Bayesian and Markov networks [91, 111, 80] where the term ‘propositional’ refers to the fact that these formalisms are extensions of basic propositional logic. As propositional logic is mostly useless describing ontologies, also these extensions are inappropriate for our setting. However, there is also a rich body of proposals for probabilistic extensions of first-order logic besides ProbFO. This class includes Markov logic, see [56, 118] and the references therein, probabilistic (deductive) databases [108, 125], inductive logic programming [115], some first-order generalizations of Bayesian networks [31], and many more; see [56, 40] for good overviews. These formalisms typically have a Herbrand-style semantics, that is, they come with a specified fixed domain and describe, in a succinct way, a fixed distribution over the set of possible worlds. The semantics given in this way enables the lifting of several efficient inference mechanisms known from propositional models, see for example [112, 39]. However, the mentioned formalisms can be viewed as being propositional, and thus, they are in a completely different spirit than Halpern *et al.*’s ProbFO which neither fixes the domain nor the set of worlds.

From a semantic and computational perspective, there is a clear similarity between ProbFO and temporal first-order logic (TFO). Both logics adopt a possible world semantics and although TFO is ‘only’ Π_1^1 -complete, just like ProbFO it is not recursively enumerable. In the case of TFO, Hodkinson, Wolter and Zakharyashev have given an elegant explanation of why this is the case and how better computational properties can be recovered, by introducing the *monodic* fragment of TFO that restricts temporal operators to be applied only to formulas with at most one free variable [78]. In fact,

monodic TFO turns out to be recursively enumerable [131] and decidable fragments of monodic TFO can often be obtained by restricting the FO part of monodic TFO to a decidable FO fragment [79, 75, 77, 76].

Contribution and Structure of the Chapter

In Section 3.1, we review syntax and semantics of ProbFO, illustrate its expressive power, and recall the computational difficulties known from [1]. We add the observation that many restrictions which to decidability of classical FO do not lead to decidability or recursive enumerability in ProbFO. In particular, we show that ProbFO is still Π_1^1 -hard when considering the two variable, monadic, or the guarded fragment. We further show that the restriction to probability values 0 and 1 leads to recursive enumerability; however, we refrain from pursuing this further as we are interested in “real” probabilistic logics, that is, we want to allow non-trivial probabilities.

In Section 3.2, we take the mentioned work on monodic temporal first-order logic as inspiration, and in the first step try to identify a monodic fragment of ProbFO. Note that the formulas of unrestricted ProbFO are obtained by combining classical FO with the language of real closed fields via real-valued terms of the form $w(\varphi)$ denoting the probability that the formula φ (with possibly free variables) is true. In analogy to TFO, a natural candidate for monodicity in ProbFO is to admit only weight terms $w(\varphi)$ in which φ has at most one free first-order variable. We show, however, that this is not an effective choice since the resulting fragment of ProbFO still fails to be recursively enumerable. We thus have to adopt stronger restrictions and define a ProbFO formula to be monodic if every weight formula contains at most one free first-order variable and no real valued variables.

Under this definition of monodicity, we establish in Section 3.3 a useful abstract representation of models of monodic ProbFO formulas—so-called *quasi-models*—which are essentially a collection of monadic formula types that satisfy certain integrity conditions and are associated with a system of polynomial inequalities over the reals to capture probabilities. The abstraction to quasi-models is the main result of this chapter and enables us to achieve the aforementioned goals for this chapter.

In Sections 3.4 and 3.5, we study the computational properties of monodic ProbFO and some of its fragments. Using quasi-models, we show in a rather direct way that the valid formulas of monodic ProbFO are recursively enumerable. Moreover, we provide a sound and complete axiomatization of monodic ProbFO. For this purpose, we extend an axiomatization of unrestricted ProbFO on finite domains of fixed size by Halpern [67] to our setting with unrestricted domains. Finally, quasi-models can be used to identify decidable fragments of monodic ProbFO. We show that for any FO-fragment \mathcal{L} such that a slightly generalized version of satisfiability in \mathcal{L} , called *realizability*, is decidable, monodic Prob \mathcal{L} is decidable, too. For the guarded fragment (GF), the two-variable fragment, and the guarded negation fragment [15], realizability is reducible to satisfiability

and thus decidable. Consequently, we obtain decidability for the case when \mathcal{L} is among the four mentioned logics. The finite model property transfers in the same way.

This decidability transfer is shown via a general algorithm, where ‘general’ refers to its applicability to all FO fragments for which realizability is decidable. Starting from that algorithm, we also analyze the computational complexity of some important decidable fragments of monodic ProbFO. The naive version of our general algorithm yields a $2\text{NEXPTIME}^{\exists\mathbb{R}, \mathcal{C}}$ upper bound where superscripts denote access to oracles. There, $\exists\mathbb{R}$ is the class of problems that reduce in polynomial time to solving systems of polynomial inequalities over the reals [121] (recall $\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}$), and \mathcal{C} is the complexity of deciding realizability in the underlying FO fragment \mathcal{L} .

As this algorithm is very general, we cannot expect tight bounds. As the next step, we propose two improvements. The first one consists of a more careful realizability check as known from monodic TFO, and this modification sometimes allows removing the oracle for \mathcal{C} . For monodic ProbGF, in particular, we obtain in this way an improved $2\text{NEXPTIME}^{\exists\mathbb{R}}$ upper bound. The second improvement is the identification of a certain model-theoretic property that we call *closure under unions of types*, and it allows improving the runtime by one exponential if \mathcal{L} satisfies it. GF satisfies the mentioned property, and thus we obtain a tight 2EXPTIME upper bound for monodic ProbGF. We also obtain a $\text{NEXPTIME}^{\exists\mathbb{R}}$ upper bound when the arity of predicates is bounded, and a tight NEXPTIME upper bound for the case where only linear weight formulas are admitted, that is, multiplication of weight terms is disallowed.

In Section 3.6, we show how monodic ProbFO can be viewed as a natural generalization of the mentioned family of probabilistic description logics. Thus, we provide a principled explanation for why these logics are computationally much more well-behaved than traditional ProbFO.

In Section 3.7, we conclude and point out interesting directions for future work.

3.1 Probabilistic First-order Logic

3.1.1 Syntax and Semantics

Let us introduce *Type-2 probabilistic first-order logic (ProbFO)* along the lines of [67]. The logic comprises two sorts: objects of the domain of discourse and the real numbers \mathbb{R} . Throughout this chapter, this is reflected by prefixing the standard FO notions with either ‘object’ or ‘field’ for referring to the respective sort. Accordingly, there are two types of variables: *object variables* and *field variables*, where the former are the standard FO variables ranging over the domain and the latter are used to represent probabilities and range over the real numbers. *Object terms* are object variables or *object constants*.

ProbFO-formulas and field terms are defined by mutual recursion:

$$\begin{aligned} \varphi, \psi &::= R(t_1, \dots, t_k) \mid \varphi \wedge \psi \mid \neg \varphi \mid \exists x \varphi(x) \mid f_1 \leq f_2 \\ f_1, f_2 &::= 0 \mid 1 \mid r \mid \mathbf{w}(\varphi) \mid f_1 + f_2 \mid f_1 \times f_2 \end{aligned}$$

where R is a k -ary predicate symbol, t_1, \dots, t_k are object terms, r is a field variable, and f_1, f_2 are field terms. Formulas of the form $f_1 \leq f_2$ are called *weight formulas*. Note that any positive integer k can be expressed as the sum $(1 + \dots + 1)$ with k summands. Moreover, rational numbers are not necessary as they can be eliminated by clearing denominators. For example $1/3 \times \mathbf{w}(A(x)) + 1/2 \times \mathbf{w}(B(x)) \leq 1/6$ is equivalent to $2 \times \mathbf{w}(A(x)) + 3 \times \mathbf{w}(B(x)) \leq 1$. Quantification $\exists x \varphi(x)$ is possible both over object and field variables x , with field variables ranging over \mathbb{R} . Moreover, we use the common abbreviations \vee, \rightarrow, \dots on first-order level and $=, <, >, \geq$ on the level of weight formulas. We use $\text{ProbFO}^=$ to denote the extension of ProbFO with equality on object terms.

As a matter of fact, the semantics of ProbFO is the same possible world semantics that is used for Prob \mathcal{ALC} as it was introduced in the preliminaries. However, a slightly different terminology is standard, so we rigorously define it again. Formulas of ProbFO are interpreted in probabilistic structures that are intuitively collections of standard FO structures (over the same domain), each carrying a weight given by a probability. More specifically, a *probabilistic structure* $\mathfrak{M} = (D, W, \mu, \pi)$ consists of a non-empty domain D , a set of worlds W , a discrete probability distribution μ over W and an *interpretation function* π that maps each pair (R, w) to a subset of D^k and each pair (c, w) to an element of D for each k -ary predicate symbol R , $w \in W$, and constant symbol c . In particular, constant symbols are interpreted in a *non-rigid* way, that is, they are not necessarily the same domain element in every world. Note that this is in contrast to Prob \mathcal{ALC} , but not a restriction; in fact, we will argue whenever necessary that our results are also valid for rigid interpretation of constants. A *valuation* for \mathfrak{M} is a function ν that maps object variables to elements of D and field variables to real numbers. Given \mathfrak{M} , ν , and a world $w \in W$, the semantics is defined similarly to standard FO:

$$\begin{aligned} (\mathfrak{M}, w, \nu) \models R(t_1, \dots, t_k) &\text{ if } (a_1, \dots, a_k) \in \pi(R, w) \text{ where} \\ &a_i \text{ is } \nu(t_i) \text{ if } t_i \text{ is a variable and } \pi(t_i, w) \text{ otherwise;} \\ (\mathfrak{M}, w, \nu) \models \neg \varphi &\text{ if } (\mathfrak{M}, w, \nu) \not\models \varphi; \\ (\mathfrak{M}, w, \nu) \models \varphi \wedge \psi &\text{ if } (\mathfrak{M}, w, \nu) \models \varphi \text{ and } (\mathfrak{M}, w, \nu) \models \psi; \\ (\mathfrak{M}, w, \nu) \models \exists x \varphi(x) &\text{ if there is } d \in D \text{ with } (\mathfrak{M}, w, \nu[x/d]) \models \varphi(x) \\ &\text{ and } x \text{ object variable;} \\ (\mathfrak{M}, w, \nu) \models \exists x \varphi(x) &\text{ if there is } d \in \mathbb{R} \text{ with } (\mathfrak{M}, w, \nu[x/d]) \models \varphi(x) \\ &\text{ and } x \text{ field variable;} \\ (\mathfrak{M}, w, \nu) \models f_1 \leq f_2 &\text{ if } [f_1]_{(\mathfrak{M}, w, \nu)} \leq [f_2]_{(\mathfrak{M}, w, \nu)}, \end{aligned}$$

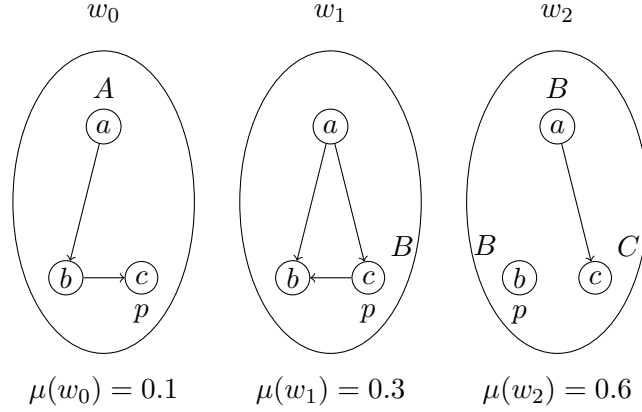


Figure 3.1: Example for a probabilistic structure.

where the interpretation $[f]_{(\mathfrak{M}, w, \nu)} \in \mathbb{R}$ of a field term f is defined in the obvious way, with terms $w(\varphi)$ interpreted as

$$[w(\varphi)]_{(\mathfrak{M}, w, \nu)} = \mu(\{w' \in W \mid (\mathfrak{M}, w', \nu) \models \varphi\}).$$

For sentences φ , we will use $\mu(\varphi)$ to abbreviate $\mu(\{w' \in W \mid (\mathfrak{M}, w', \nu) \models \varphi\})$ for an arbitrary (not relevant) ν , when \mathfrak{M} is clear.

A ProbFO-sentence φ is *satisfiable* if there is a probabilistic structure $\mathfrak{M} = (D, W, \mu, \pi)$ and a world $w \in W$ such that $(\mathfrak{M}, w) \models \varphi$. In such a case, we also write $\mathfrak{M} \models \varphi$. A sentence φ is *valid* if $\neg\varphi$ is not satisfiable. Note that the world witnessing satisfiability can have weight 0; this might be undesirable in potential AI applications where one is interested in satisfiability of a formula φ in worlds with positive probability. However, one can check satisfiability of $w(\varphi) > 0$ in this case. Let us illustrate the introduced concepts using an example.

Example 3.1. An example for a probabilistic structure $\mathfrak{M} = (D, W, \mu, \pi)$ is depicted in Figure 3.1, where possible worlds w_0, w_1, w_2 are indicated by balloons containing standard first-order relational structures, that is, dots indicate domain elements, capital letters indicate the interpretation of unary predicate symbols and edges indicate the interpretation of a binary predicate symbol. The weight of each world is shown below the according world, and p denotes the interpretation of a constant name p . Formally, the depicted probabilistic structure is as follows:

- $D = \{a, b, c\}$;
- $W = \{w_0, w_1, w_2\}$ with $\mu(w_0) = 0.1$, $\mu(w_1) = 0.3$, $\mu(w_2) = 0.6$ whose sum is 1;

- $\pi(A, w_0) = \{a\}$, $\pi(A, w_1) = \pi(A, w_2) = \emptyset$;
- $\pi(B, w_0) = \emptyset$, $\pi(B, w_1) = \{c\}$, $\pi(B, w_2) = \{a, b\}$;
- $\pi(C, w_0) = \pi(C, w_1) = \emptyset$, $\pi(C, w_2) = \{c\}$;
- $\pi(R, w_0) = \{(a, b), (b, c)\}$, $\pi(R, w_1) = \{(a, b), (a, c), (c, b)\}$, $\pi(R, w_2) = \{(a, c)\}$.
- $\pi(p, w_0) = c$, $\pi(p, w_1) = c$, $\pi(p, w_2) = b$.

Consider the formulas

$$\begin{aligned} \varphi_1 &= \forall x \mathbf{w}(B(x)) \geq 0.1; & \varphi_2 &= \mathbf{w}(\forall x B(x)) \geq 0.1; \\ \varphi_3 &= \mathbf{w}(B(p)) = 0.9; & \varphi_4 &= \exists xy B(x) \wedge C(y) \wedge \mathbf{w}(R(x, y)) \geq 0.5; \\ \varphi_5 &= \forall xy (B(x) \wedge C(y)) \rightarrow \mathbf{w}(R(x, y)) \geq 0.5. \end{aligned}$$

It is not hard to verify that $(\mathfrak{M}, w) \models \varphi_1$ for all worlds $w \in W$ since every domain element satisfies B in some world with positive probability. In contrast, $(\mathfrak{M}, w) \not\models \varphi_2$ for every $w \in W$, since there is no world where B is satisfied for all elements. φ_3 is satisfied, as the constant p satisfies B in worlds w_1 and w_2 whose weights sum up to 0.9. Further, we have $(\mathfrak{M}, w_2) \models \varphi_4$ which is witnessed by valuation ν with $\nu(x) = a$ and $\nu(y) = c$. In contrast, $(\mathfrak{M}, w_2) \not\models \varphi_5$ since we can choose $\nu(x) = b$ and $\nu(y) = c$. However, we have $(\mathfrak{M}, w_1) \models \varphi_5$.

Sometimes, we want to *abstract* from the weight formulas and view a ProbFO formula as a plain first-order formula. For this purpose, we denote with $\overline{\varphi}$ the FO formula that is obtained from the ProbFO formula φ by replacing each weight formula $f_1 \leq f_2$ that is not within the scope of another weight formula and has k free variables x_1, \dots, x_k with $P_{f_1 \leq f_2}(x_1, \dots, x_k)$, where $P_{f_1 \leq f_2}$ is a fresh k -ary predicate symbol. As an example, φ_1 and φ_4 are abstracted to

$$\overline{\varphi_1} = \forall x P_{\mathbf{w}(B(x)) \geq 0.1}(x) \quad \text{and} \quad \overline{\varphi_4} = \exists xy (B(x) \wedge C(y) \wedge P_{\mathbf{w}(R(x, y)) \geq 0.5}(x, y)).$$

This notation is lifted to sets of formulas in the obvious way. Note that this is an abstraction in the sense that $\overline{\varphi}$ might be satisfiable, while φ is unsatisfiable. Conversely, however, unsatisfiability (resp., validity) of $\overline{\varphi}$ implies unsatisfiability (resp., validity) of φ .

3.1.2 Examples and Expressivity

Let us demonstrate the expressive power of ProbFO using a small example. This example is orthogonal to the possible applications of ProbFO in biomedical domains. It illustrates some typical design choices for writing a ProbFO-ontology and presents instances for reasoning in ProbFO.

Example 3.2 (Street food). *Imagine you are traveling in India and are offered food in the street. You recall that your travel guide warned about street food, which is why you believe that you probably not tolerate the offered food. Your belief is represented as the following ProbFO sentence:*

$$\forall x \, w(\text{tolerate}(l, x) \mid \text{streetfood}(x)) \leq 0.1 \quad (3.1)$$

where l is a constant representing yourself and where we encode statements about conditional probabilities by multiplying out denominators, following Halpern [67]: $w(\varphi \mid \psi) \geq p$ abbreviates $w(\varphi \wedge \psi) \geq p \times w(\psi)$. Your travel guide additionally says that local Indian might be used to the present hygienic conditions. Thus, the belief about your own tolerance does not generalize and you represent this as follows:

$$\exists y \forall x \, w(\text{tolerate}(y, x) \mid \text{streetfood}(x) \wedge \text{Indian}(y)) \geq 0.9. \quad (3.2)$$

Despite all the warnings and your skepticism, you decide to at least have a look at the food. It turns out that it looks delicious and smells good, so you are inclined to eat it. However, you consider again your internal knowledge base consisting of the above two sentences (3.1) and (3.2). In particular, you ask whether it implies anything about

$$w(\text{tolerate}(l, \text{curry}) \mid \text{streetfood}(\text{curry}) \wedge \text{looksGood}(\text{curry})),$$

where curry is a constant representing the dish you see. A quick calculation shows that you are ignorant about things that are street food and look good, that is, you ‘infer’ the trivial consequence that the above probability is in $[0, 1]$.

Thus, you become insecure, and consult the travel guide again for more specific information. You find a sentence saying that “street food should be avoided even if the vendors tell you that it is OK”. You understand that this is a general warning, and conclude that the look/smell/promotion/... of street food is independent from how well you probably tolerate it. You represent this as the conditional independence

$$\forall x \, \text{indep}(\text{tolerate}(I, x), \text{looksGood}(x) \mid \text{streetfood}(x)) \quad (3.3)$$

for which you add the following sentence to your knowledge base:

$$\forall x \, w(\text{tolerate}(I, x) \mid \text{streetfood}(x) \wedge \text{looksGood}(x)) = w(\text{tolerate}(I, x) \mid \text{streetfood}(x)).$$

Hence, you finally conclude that you probably should not eat the curry .

Thus, in ProbFO we are able to describe situations involving uncertainty about the environment that you or a potential agent might face. In particular, it has means to express independence statements such as formula (3.3). Consequently, ProbFO encompasses standard models for probabilistic reasoning like Bayesian networks, and can, moreover, express also sets of (in-)dependences that cannot be encoded by any Bayesian

network. Note, however, that Bayesian networks are more succinct a representation, as they can encode all (in-)dependencies in a domain (exponentially many!) in a potentially small graph; please consult [111] for an overview. The purpose of ProbFO is orthogonal: it is based on the open-world assumption, thus we do *not* need to completely specify all conditional probabilities and we can express substantially more using the included first-order logic.

3.1.3 Complexity and first Observations

The high expressive power of ProbFO comes at the price of high computational complexity. In the following, we are interested in the complexity of checking validity. As ProbFO contains FO, it is certainly undecidable and at best recursively enumerable. For full ProbFO, however, the complexity results have been discouraging. More specifically, Abadi and Halpern have shown that validity in ProbFO is Π_1^2 -complete and, thus, highly undecidable and far from being recursively enumerable [1]. Trying to pinpoint the source of the high complexity they considered fragments with restricted vocabularies. Surprisingly, they show that already over vocabularies that contain *only constants*, validity is Π_∞^1 -complete when equality is allowed. The lower bounds of these theorems are proved by reductions from suitable higher-order theories of integer arithmetics and extensively use quantification over real variables. On the positive side, Halpern provided a sound and complete axiomatization of ProbFO when the domains are bounded by a constant N [67].

We give additional evidence of the computational difficulty of ProbFO by considering some standard approaches to get decidability of satisfiability in classical FO. More precisely, we prohibit quantification over real variables and study the following fragments of ProbFO:

- the two-variable fragment, where at most two (object) variables are allowed;
- the monadic fragment, where only unary predicate symbols are allowed;
- the guarded fragment, which is defined in analogy of guarded temporal first-order logic [78]: A ProbFO formula φ is guarded, when the FO-abstraction $\bar{\psi}$ of each subformula ψ of φ is guarded;
- the guarded negation fragment, similarly to the guarded fragment: A ProbFO formula φ has guarded negation, if the FO-abstraction $\bar{\psi}$ of every subformula ψ of φ has guarded negation.

Thus, for example, the formula

$$\exists xy (\mathbf{w}(A(x)) + \mathbf{w}(A(y)) \geq 1 \wedge P(x) \wedge Q(y))$$

is guarded, the guard being the atom $w(A(x)) + w(A(y)) \geq 1$. We prove that for all these fragments, validity is not recursively enumerable. The proof is by a reduction of recurring domino systems that is rather different in spirit from the mentioned reductions from integer arithmetic and thus provides additional intuition about the hardness of ProbFO.

Theorem 3.3. *Validity in ProbFO is Π_1^1 -hard for*

- *the monadic, two-variable fragment of ProbFO and*
- *the guarded fragment (and thus the guarded negation fragment) of ProbFO,*

even if quantification over field variables is disallowed.

Proof. We give the full proof only for the case of two object variables and sketch how to adapt it to the monadic and the guarded case. The proof is via a reduction from recurring tiling problems which are known to be Σ_1^1 -hard [72]. A *recurring tiling problem* is a quadruple $P = (T, H, V, t_r)$, where T is a finite set of *tile types*, $H, V \subseteq T \times T$ are the *horizontal and vertical matching conditions*, and $t_r \in T$ is the *recurrent tile*. A *solution* to P is a mapping $\tau : \mathbb{N} \times \mathbb{N} \rightarrow T$ such that

- $(\tau(i, j), \tau(i, j + 1)) \in H$ for all $i, j \geq 0$;
- $(\tau(i, j), \tau(i + 1, j)) \in V$ for all $i, j \geq 0$;
- there are infinitely many $j \geq 0$ such that $\tau(0, j) = t_r$

Let P be a tiling problem as above. We use the FO dimension to represent the vertical dimension of the grid, and possible worlds for the horizontal dimension. To represent the successor relation in the vertical direction, we introduce a binary relation R . Of course, R should be unbounded and rigid, there should be exactly one tile type at every grid element, and the tiling should be compatible with vertical successors:

$$w(\forall x \exists y R(x, y)) = 1 \tag{3.4}$$

$$\forall x \forall y R(x, y) \Rightarrow w(R(x, y)) = 1 \tag{3.5}$$

$$w(\forall x (\bigvee_{t \in T} X_t(x) \wedge \bigwedge_{t, t' \in T, t \neq t'} \neg(X_t(x) \wedge X_{t'}(x)))) = 1 \tag{3.6}$$

$$w(\forall x \forall y (R(x, y) \Rightarrow \bigvee_{(t, t') \in V} X_t(x) \wedge X_{t'}(y))) = 1. \tag{3.7}$$

To represent the successor relation in the horizontal direction, we use probabilities. More precisely, a grid node in row i is represented by a domain element that satisfies A with

probability $1/2^i$. To make this work, we first enforce that the probability of any element to satisfy A is $1/2^i$ for some $i \in \mathbb{N}$, and that all probabilities of this form are present:

$$\forall x (\mathbf{w}(A(x)) = 1 \vee \exists y (\mathbf{w}(A(y)) = 2\mathbf{w}(A(x)))) \quad (3.8)$$

$$\forall x \exists y 2\mathbf{w}(A(y)) = \mathbf{w}(A(x)). \quad (3.9)$$

These probabilities, though, are still associated with the FO dimension. To transfer our probability scheme to the dimension of possible worlds, we force that in every world, there is at least one element that satisfies the unary predicate M and all elements that satisfy M agree on the probability of satisfying A ; moreover, every element is marked in at least one world:

$$\mathbf{w}(\exists x M(x)) = 1 \quad (3.10)$$

$$\mathbf{w}(\forall x \forall y ((M(x) \wedge M(y)) \Rightarrow \mathbf{w}(A(x)) = \mathbf{w}(A(y)))) = 1 \quad (3.11)$$

$$\forall x \mathbf{w}(M(x)) > 0. \quad (3.12)$$

In this way, every *world* is associated with a unique probability: the probability of the M -marked elements to satisfy A ; conversely, for each probability $p = 1/2^i$, there is a world associated with p . Note that the probabilities associated with worlds in the described way are *not* the probabilities that are assigned to worlds by a probabilistic structure; in particular, the probabilities associated with worlds need not sum up to one. We can now enforce as follows that the tiling is compatible also with horizontal successors:

$$\mathbf{w}\left(\bigwedge_{t \in T} \forall x X_t(x) \Rightarrow \bigvee_{(t,t') \in H} (\exists y M(y) \wedge \mathbf{w}(\psi_{t'}(x, y)) = 1)\right) = 1 \quad (3.13)$$

where

$$\psi_{t'}(x, y) = \exists x (M(x) \wedge \mathbf{w}(A(y)) = 2\mathbf{w}(A(x))) \Rightarrow X_{t'}(x).$$

It remains to enforce that the recurring tile t_r occurs infinitely often. We first introduce a new unary predicate symbol C_0 that marks the first row and ensure that, in this column, the recurring tile t_r occurs at least once:

$$\exists x (\mathbf{w}(C_0(x)) = 1 \wedge \mathbf{w}(X_{t_r}(x)) > 0). \quad (3.14)$$

Now infinite occurrence of t_r can be expressed as follows:

$$\begin{aligned} &\mathbf{w}(\forall x ((C_0(x) \wedge X_{t_r}(x)) \Rightarrow \\ &\quad \exists y (\exists x (M(x) \wedge \mathbf{w}(A(y)) < \mathbf{w}(A(x))) \wedge \mathbf{w}(\vartheta(x, y)) = 1))) = 1 \end{aligned} \quad (3.15)$$

where

$$\vartheta(x, y) = \exists x (M(x) \wedge \mathbf{w}(A(y)) = \mathbf{w}(A(x))) \Rightarrow X_{t_r}(x).$$

Let φ_P be the conjunction of all ProbFO sentences above. It remains to show the following

Claim. P has a solution iff φ_P is satisfiable.

Proof of the Claim. For the “if”-direction, let $\mathfrak{M} = (D, W, \mu, \pi)$ be a probabilistic structure that satisfies φ_P . By Formula (3.8), there is some $e \in D$ satisfying A with probability 1. We say that a world w *associated with probability* p if all domain elements from $\pi(M, w)$ satisfy A with probability p . Observe that this is well-defined due to Formula (3.11). Moreover, due to Formula (3.10), every world is associated with a probability and by Formulas (3.8) and (3.9), for every probability p of the form $1/2^i$, there is a world associated with p , and all worlds are associated with a probability of this form.

By Formula (3.12), there is some world w_0 such that $e \in \pi(M, w_0)$, thus the probability associated with w_0 is 1. Starting at w_0 , fix an infinite sequence of worlds w_0, w_1, w_2, \dots such that the associated probability of w_i is $1/2^i$. Note that this is possible by what was said above.

By Formula (3.14), there is some $d_0 \in D$ that satisfies C_0 with probability 1 and X_{t_r} in some world. Due to Formula (3.15), d_0 satisfies X_{t_r} in infinitely many worlds among w_0, w_1, \dots . Starting at d_0 , fix an infinite sequence of elements d_0, d_1, d_2, \dots such that d_i is related to d_{i+1} by R , for all $i \geq 0$ and in all worlds $w \in W$. This is enabled by Formulas (3.4) and (3.5). We can read off a mapping $\tau : \mathbb{N} \times \mathbb{N} \rightarrow T$ as follows:

$$\tau(i, j) = t \quad \Leftrightarrow \quad d_i \in \pi(X_t, w_j).$$

The mapping τ is well-defined since every domain element satisfies in every world X_t for precisely one tile type $t \in T$, by Formula (3.6). Moreover, the vertical matching condition is satisfied due to (3.7). Finally observe that the horizontal matching condition is satisfied because of (3.13).

For the “only if”-direction, assume that P has a solution τ . We define a probabilistic structure $\mathfrak{M} = (D, W, \mu, \pi)$ by taking $D = W = \mathbb{N}$ and

$$i \in \pi(X_t, j) \quad \Leftrightarrow \quad \tau(i, j) = t \quad \text{for all } i, j \in \mathbb{N}, t \in T.$$

That is, domain elements model the vertical dimension and worlds the horizontal dimension of the tiling. It remains to give the interpretation of the predicates C_0 , R , A , and M , and ensure that world i is associated with probability $1/2^i$, in the sense explained above. Specifically, we have for all $j \in W$:

$$\begin{aligned} \mu(j) &= 1/2^{j+1}; \\ \pi(R, j) &= \{(i, i+1) \mid i \in \mathbb{N}\}; \\ \pi(A, j) &= \{i \mid i \leq j\}; \\ \pi(M, j) &= \{j\}; \\ \pi(C_0, j) &= \{0\}. \end{aligned}$$

It is now not hard to verify that $\mathfrak{M} \models \varphi_P$, which finishes the proof of the claim.

Note that φ_P is a formula in the two-variable fragment, however, it still uses the binary predicate symbol R . For getting rid of this predicate, we introduce fresh unary symbols P, Q and replace $R(x, y)$ with $w(P(x) \wedge Q(y)) > 0$ in Formulas (3.4), (3.5), and (3.7). Note that under this modification, Formula (3.5) becomes a tautology and can be omitted. It is straightforward to adapt the above proof. This establishes the first point of the theorem.

For the guarded fragment, observe that all formulas, except for (3.11), are guarded according to our definition: they contain a formula of the form $w(A(x)) = 2w(A(y))$ which abbreviates $w(A(x)) \leq 2w(A(y)) \wedge w(A(x)) \geq 2w(A(y))$ one of whose atoms can be used as a guard. Formula (3.11) can, however, be rewritten as follows:

$$w(\forall x \forall y (w(A(x)) \leq w(A(y)) \Rightarrow (w(A(x)) \geq w(A(y)) \vee \neg M(x) \vee \neg M(y)))) = 1 \quad (3.11^*)$$

This finishes the proof of the theorem. ■

A standard approach to lower the computational complexity of probabilistic logics has been to restrict weight formulas to the form $w(\varphi) > 0$ or $w(\varphi) = 1$ [101, 24]. It turns out that this restriction indeed also leads to recursive enumerability in ProbFO, even when equality is allowed, and thus it fulfills our first requirement. However, one can hardly talk about a *probabilistic* logic anymore, which is why we will later concentrate on a different approach.

We denote with $\text{ProbFO}_{01}^=$ the fragment of $\text{ProbFO}^=$ which allows only for weight formulas of the form $w(\varphi(\vec{x})) > 0$ and $w(\varphi(\vec{x})) = 1$ and show recursive enumerability by a satisfiability preserving reduction to $\text{FO}^=$. Intuitively, a domain element in a first-order structure will correspond to a *pair* (d, w) of domain element and world in a probabilistic structure, and two fresh binary predicate symbols D and W are used to simulate accessibility inside a world and among worlds, respectively. More precisely, $(a, b) \in \pi(D)$ means that a and b represent the same domain element in different worlds, and $(a, b) \in \pi(W)$ means that a and b refer to different domain elements in the same world.

Theorem 3.4. *Validity in $\text{ProbFO}_{01}^=$ is recursively enumerable.*

Proof. The proof is provided by giving a satisfiability preserving reduction to $\text{FO}^=$. For the sake of simplicity, we consider satisfiability in worlds with probability greater than 0; this is without loss of generality, which can be proved along the lines of the proof of Lemma 4.2 in the next chapter.

Let φ be a $\text{ProbFO}_{01}^=$ sentence. As a preliminary step, we show that we can assume without loss of generality that φ is constant free. For this purpose, assume that $\{c_1, \dots, c_m\}$ is the set of all constants appearing in φ . We simulate them using fresh unary predicate symbols C_i , $1 \leq i \leq m$. More precisely, we obtain a formula φ' by replacing

every atom $P(t_1, \dots, t_n)$ in φ with the formula $\exists z_1 \dots \exists z_m P(t'_1, \dots, t'_n) \wedge \bigwedge_{i=1}^m C_i(z_i)$, where

$$t'_i = \begin{cases} t_i & \text{if } t_i \text{ is variable;} \\ z_j & \text{if } t_i \text{ is the constant } c_j. \end{cases}$$

Moreover, define a formula ψ expressing that in every world there is precisely one element satisfying C_i for every $1 \leq i \leq m$ (recall that we work with non-rigid constants):

$$\psi = \mathbf{w}(\exists z_1 \dots \exists z_m \bigwedge_{i=1}^m C_i(z_i) \wedge \forall x C_i(x) \rightarrow x = z_i) = 1.$$

Using the definition of ψ , it is straightforward to verify the following Claim.

Claim 1. φ is satisfiable iff $\varphi' \wedge \psi$ is satisfiable.

Thus, we can assume that φ is constant free. We give a satisfiability preserving reduction to $\text{FO}^=$. For this purpose, introduce two fresh predicate symbols D and W simulating accessibility inside a world and among worlds. Define a sentence ψ_1 as the conjunction of the following:

- sentences enforcing both D and W to be equivalence relations;
- sentences for left commutativity, right commutativity, and Church-Rosser property, respectively, that is for axiomatizing product structures, see [52]:

$$\begin{aligned} \text{lcom}(D, W) &= \forall x \forall y \forall z (D(x, y) \wedge W(y, z) \rightarrow \exists u W(x, u) \wedge D(u, z)); \\ \text{rcom}(D, W) &= \forall x \forall z \forall y (W(x, y) \wedge D(y, z) \rightarrow \exists u D(x, u) \wedge W(u, z)); \\ \text{cr}(D, W) &= \forall x \forall y \forall z (D(x, y) \wedge W(x, z) \rightarrow \exists u W(y, u) \wedge D(z, u)); \end{aligned}$$

- a sentence expressing that equivalence classes of D and W intersect in at most one element:

$$\forall x \forall y W(x, y) \wedge D(x, y) \rightarrow x = y;$$

- a sentence enforcing that predicate symbols in φ are interpreted inside a world, that is, for each k -ary symbol R the following:

$$\forall x_1 \dots \forall x_k (R(x_1, \dots, x_k) \rightarrow (W(x_1, x_2) \wedge \dots \wedge W(x_{k-1}, x_k))).$$

Intuitively, every equivalence class of D represents a domain element and every equivalence class of W represents a world. We need a further fresh unary predicate symbol World for defining a unique representative for every world. This is expressed by the following formula ψ_2 :

$$\psi_2 = (\forall x \exists y W(x, y) \wedge \text{World}(y)) \wedge (\forall x y W(x, y) \wedge \text{World}(x) \wedge \text{World}(y) \rightarrow x = y).$$

It remains to define the translation function from ProbFO_{01}^- to FO^- . We define the function fo_w parametrized by a world variable w , that intuitively stores the world where we are evaluating, as follows:

$$\begin{aligned}
 \text{fo}_w(P(x_1, \dots, x_k)) &= P(x_1, \dots, x_k) \\
 \text{fo}_w(x = y) &= (x = y) \\
 \text{fo}_w(\neg\psi) &= \neg\text{fo}_w(\psi) \\
 \text{fo}_w(\psi_1 \wedge \psi_2) &= \text{fo}_w(\psi_1) \wedge \text{fo}_w(\psi_2) \\
 \text{fo}_w(\exists x \psi(x, y_1, \dots, y_k)) &= \exists x \mathbf{W}(x, w) \wedge \text{fo}_w(\psi(x, y_1, \dots, y_k)) \\
 \text{fo}_w((\mathbf{w}(\psi(y_1, \dots, y_k)) > 0)) &= \exists w' \exists y'_1 \dots \exists y'_k (\mathbf{World}(w') \wedge \bigwedge_{i=1}^k \mathbf{W}(w', y'_i) \wedge \mathbf{D}(y_i, y'_i) \\
 &\quad \wedge \text{fo}_{w'}(\psi(y'_1, \dots, y'_k)))
 \end{aligned}$$

Claim 2. φ is satisfiable iff $\psi_1 \wedge \psi_2 \wedge \exists w (\mathbf{World}(w) \wedge \text{fo}_w(\varphi))$ is satisfiable.

Proof of Claim 2. For the “ \Rightarrow ”-direction, assume a model $\mathfrak{M} = (D, W, \mu, \pi)$ of φ and define the first-order structure $\mathfrak{A} = (A, \sigma)$ as follows:

- $A = D \times W$;
- $\sigma(P) = \{((d_1, w), \dots, (d_k, w)) \mid w \in W, (d_1, \dots, d_k) \in \pi(P, w)\}$ for all predicate symbols;
- $\sigma(\mathbf{D}) = \{((d, w), (d', w)) \mid d, d' \in D, w \in W\}$;
- $\sigma(\mathbf{W}) = \{((d, w), (d, w')) \mid d \in D, w, w' \in W\}$;
- $\sigma(\mathbf{World}) = \{(d_0, w) \mid w \in W\}$ for a fixed $d_0 \in D$.

It is routine to verify that $\mathfrak{A} \models \psi_1 \wedge \psi_2$. By structural induction on ψ , we show that the following equivalence holds for all formulas ProbFO_{01}^- formulas ψ , all worlds $w \in W$, and all valuations ν

$$(\mathfrak{M}, w, \nu) \models \psi(y_1, \dots, y_k) \Leftrightarrow (\mathfrak{A}, \nu_w) \models \text{fo}_w(\psi(y_1, \dots, y_k)) \quad (3.16)$$

where $\nu_w(x) = (\nu(x), w)$ for all variables x in ψ and $\nu_w(w) = (d_0, w)$. The proof of this equivalence is not hard and we give only the case when $\psi = \mathbf{w}(\psi'(y_1, \dots, y_k)) > 0$.

“ \Rightarrow ”: If $(\mathfrak{M}, w, \nu) \models \mathbf{w}(\psi'(y_1, \dots, y_k)) > 0$, there is a world w' such that $(\mathfrak{M}, w', \nu) \models \psi'(y_1, \dots, y_k)$. By induction, we have $(\mathfrak{A}, \nu_{w'}) \models \text{fo}_{w'}(\psi'(y_1, \dots, y_k))$. By definition of $\sigma(\mathbf{D})$, $\sigma(\mathbf{W})$, $\sigma(\mathbf{World})$, and $\nu_{w'}$, we get $(\mathfrak{A}, \nu_w) \models \text{fo}_w(\mathbf{w}(\psi'(y_1, \dots, y_k)) > 0)$.

“ \Leftarrow ”: If $(\mathfrak{A}, \nu_w) \models \text{fo}_w(\mathbf{w}(\psi'(y_1, \dots, y_k)) > 0)$, there are domain elements a, b_1, \dots, b_k with $a \in \sigma(\text{World})$, $(a, b_i) \in \sigma(W)$ and $(b_i, \nu_w(y_i)) \in \sigma(D)$ for all $1 \leq i \leq k$ such that $(\mathfrak{A}, \widehat{\nu_w}) \models \text{fo}_{w'}(\psi'(y'_1, \dots, y'_k))$ where

$$\widehat{\nu_w} = \nu_w[w' \rightarrow a, y'_1 \rightarrow b_1, \dots, y'_k \rightarrow b_k].$$

As w', y'_1, \dots, y'_k are all free variables in ψ' , we also have $(\mathfrak{A}, \overline{\nu_w}) \models \text{fo}_{w'}(\psi'(y_1, \dots, y_k))$ where

$$\overline{\nu_w} = \nu_w[w' \rightarrow a, y_1 \rightarrow b_1, \dots, y_k \rightarrow b_k].$$

Since $a \in \sigma(\text{World})$, $a = (d_0, w')$ for some $w' \in W$. Hence, each b_i is of the form (d_i, w') for some $d_i \in D$. Thus, $\overline{\nu_w} = \nu_{w'}$ and the induction hypothesis yields $(\mathfrak{M}, w', \nu) \models \psi'(y_1, \dots, y_k)$. This finally implies $(\mathfrak{M}, w, \nu) \models \mathbf{w}(\psi'(y_1, \dots, y_k)) > 0$ and finishes the proof of (3.16).

It remains to note that (3.16) applied to φ yields $\mathfrak{A} \models \exists w (\text{World}(w) \wedge \text{fo}_w(\varphi))$.

“ \Leftarrow ”: Assume a model $\mathfrak{A} = (A, \sigma)$ of $\psi_1 \wedge \psi_2 \wedge \exists w (\text{World}(w) \wedge \text{fo}_w(\varphi))$. We define a probabilistic structure $\mathfrak{M} = (D, W, \nu, \pi)$ as follows:

- D is the set of equivalence classes of $\sigma(D)$, that is, each $d \in D$ is a subset of A ;
- W is the set of equivalence classes of $\sigma(W)$, that is, each $w \in W$ is a subset of A ;
- $\pi(P, w) = \{(d_1, \dots, d_k) \mid (e_1, \dots, e_k) \in \sigma(P), d_i \cap w = \{e_i\} \text{ for all } 1 \leq i \leq k\}$;
- μ is an arbitrary discrete probability distribution over W (by the Löwenheim-Skolem Theorems we can assume that A (and thus W) is countable; hence, such a distribution exists).

Note that \mathfrak{M} is well-defined since D and W satisfy ψ_1 ; in particular, $d \cap w$ is a singleton set for all $d \in D$ and $w \in W$ and predicate symbols are only interpreted inside worlds. We show that \mathfrak{M} satisfies the following for all ProbFO $_{01}^-$ formulas $\psi(y_1, \dots, y_k)$, worlds $w \in W$, and valuations ν

$$(\mathfrak{M}, w, \nu) \models \psi(y_1, \dots, y_k) \iff (\mathfrak{A}, \nu_w) \models \text{fo}_w(\psi(y_1, \dots, y_k)) \quad (3.17)$$

where $\nu_w(x)$ is the unique element in $\nu(x) \cap w$ for all variables x in ψ and $\nu_w(w)$ is the unique $v \in w \cup \sigma(\text{World})$, which is well-defined due to ψ_2 . Again, the proof of Equation (3.17) is straightforward and we show only the case when $\psi = \mathbf{w}(\psi'(y_1, \dots, y_k)) > 0$.

“ \Rightarrow ”: If $(\mathfrak{M}, w, \nu) \models \mathbf{w}(\psi'(y_1, \dots, y_k)) > 0$, there is a world w' such that $(\mathfrak{M}, w', \nu) \models \psi'(y_1, \dots, y_k)$. By induction, we have $(\mathfrak{A}, \nu_{w'}) \models \text{fo}_{w'}(\psi'(y_1, \dots, y_k))$. By definition of $\sigma(D)$, $\sigma(W)$, $\sigma(\text{World})$, and $\nu_{w'}$, we get $(\mathfrak{A}, \nu_w) \models \text{fo}_w(\mathbf{w}(\psi'(y_1, \dots, y_k)) > 0)$.

“ \Leftarrow ”: If $(\mathfrak{A}, \nu_w) \models \text{fo}_w(\mathbf{w}(\psi'(y_1, \dots, y_k)) > 0)$, there are domain elements a, b_1, \dots, b_k with $a \in \sigma(\text{World})$, $(a, b_i) \in \sigma(W)$ and $(b_i, \nu_w(y_i)) \in \sigma(D)$ for all $1 \leq i \leq k$ such that $(\mathfrak{A}, \widehat{\nu_w}) \models \text{fo}_{w'}(\psi'(y'_1, \dots, y'_k))$ where

$$\widehat{\nu_w} = \nu_w[w' \rightarrow a, y'_1 \rightarrow b_1, \dots, y'_k \rightarrow b_k].$$

As w', y'_1, \dots, y'_k are all free variables in ψ' , we also have $(\mathfrak{A}, \overline{\nu_w}) \models \text{fo}_{w'}(\psi'(y_1, \dots, y_k))$ where

$$\overline{\nu_w} = \nu_w[w' \rightarrow a, y_1 \rightarrow b_1, \dots, y_k \rightarrow b_k].$$

Put $w_a = [a]_{\mathbb{W}}$ and $d_i = [b_i]_{\mathbb{D}}$ for all $1 \leq i \leq k$.¹ Note that $w_a \cap d_i = \{b_i\}$ for every i and a is the unique element from w_a in $\sigma(\text{World})$. In particular, $\overline{\nu_w} = \nu_{w'}$ and the induction hypothesis yields $(\mathfrak{M}, w', \nu) \models \psi'(y_1, \dots, y_k)$. This finally implies $(\mathfrak{M}, w, \nu) \models \text{w}(\psi'(y_1, \dots, y_k)) > 0$ and finishes the proof of (3.17).

By assumption, we have that $\mathfrak{A} \models \exists w (\text{World}(w) \wedge \text{fo}_w(\varphi))$, hence $(\mathfrak{A}, \nu') \models \text{fo}_w(\varphi)$ with $\nu'(w) = a$ for some $a \in \sigma(\text{World})$. Note that we have $\nu' = \nu_{[a]_{\mathbb{W}}}$ where ν is the empty valuation. Thus, Equation (3.17) yields $(\mathfrak{M}, [a]_{\mathbb{W}}, \nu) \models \varphi$. ■

Let us finish this section by noting that an analogous reduction is possible when rigid constants are assumed. In particular, we define a formula

$$\psi_3 = \forall x \forall y \text{D}(x, y) \Rightarrow \left(\bigwedge_{i=1}^k C_i(x) \Leftrightarrow C_i(y) \right)$$

which intuitively expresses that any two individuals representing the same domain element satisfy the same predicates C_i (recall that the C_i mimic the use of the constant symbols). One can then show along the lines of the above proof that φ is satisfiable iff $\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \exists w (\text{World}(w) \wedge \text{fo}_w(\varphi))$ is satisfiable.

3.2 Monodic ProbFO

The complexity results in the previous section, particularly Theorem 3.3, illustrate that several restrictions of ProbFO that might seem promising on first sight fail to improve the computational properties of this logic. Inspired by the good computational properties of *monodic fragments* of temporal first-order logic [78, 131], we aim to define monodic fragments of ProbFO that are computationally well-behaved. In the context of temporal first-order logic, a formula is monodic when temporal operators are applied only to formulas with at most one free variable. We first show that one has to be careful when adapting this notion to ProbFO.

Theorem 3.5. *Validity in ProbFO is Π_1^0 -hard (i.e., not recursively enumerable) even if only one free object variable is allowed to occur in weight formulas.*

Proof. The proof is by a reduction from finite validity in FO which is not recursively enumerable. Let φ be an FO sentence, take a fresh unary predicate P , and start with

¹As usual, we denote with $[a]_R$ the equivalence class of a in an equivalence relation R .

enforcing that for every domain element, the probability to satisfy P is $1/2^i$ for some $i \in \mathbb{N}$:

$$\forall x \forall r \left(\mathbf{w}(P(x)) = r \Rightarrow \exists y \left(r = 1 \vee \mathbf{w}(P(y)) = 2r \right) \right).$$

Next, guarantee that there are no infinite decreasing chains and thus only finitely many probabilities of satisfying P actually occur:

$$\exists r (r > 0 \wedge \forall y \mathbf{w}(P(y)) \geq r).$$

Note that there can still be infinitely many elements with identical probabilities of satisfying P . We cannot prevent this, but we can force that ‘having the same probability of satisfying P ’ is a congruence regarding all relations that occur in φ . We illustrate this for a unary predicate A and a binary predicate R :

$$\begin{aligned} & \forall r \forall x \forall y \left((A(x) \wedge \mathbf{w}(P(x)) = r \wedge \mathbf{w}(P(y)) = r) \Rightarrow A(y) \right); \\ & \forall r_1 \forall r_2 \forall x_1 \forall x_2 \forall y_1 \forall y_2 \left((R(x_1, y_1) \wedge \mathbf{w}(P(x_1)) = r_1 \wedge \mathbf{w}(P(x_2)) = r_1 \wedge \right. \\ & \quad \left. \mathbf{w}(P(y_1)) = r_2 \wedge \mathbf{w}(P(y_2)) = r_2) \Rightarrow R(x_2, y_2) \right). \end{aligned}$$

Let ψ be the conjunction of the above ProbFO formulas. It is now easy to see that φ is finitely valid iff $(\psi \Rightarrow \varphi)$ is a ProbFO validity. ■

The proof nicely illustrates why the restriction formulated in Theorem 3.5 – although a natural candidate for monodicity – is not strong enough to regain recursive enumerability. By quantifying over field variables, it is still possible to compare the probabilities of different domain elements, which is precisely what the monodicity condition seeks to prevent. To avoid this, we strengthen monodicity and require that weight formula with a free object variable have *no* other free variable (object or field). This restriction makes field variables and quantification over them mostly useless regarding the possibility to talk about probabilities, so we disallow them altogether.

Definition 3.6 (Monodic ProbFO formula). *A ProbFO formula is monodic if it contains no field variables and every weight formula contains at most one free (object) variable.*

3.2.1 Examples and Expressivity

We will see that the above definition of monodicity indeed guarantees good computational properties such as recursive enumerability of validity in monodic ProbFO. Of course, in the balance we lose expressive power to some degree. As discussed before Definition 3.6, monodicity *syntactically* restricts the ability to relate different domain elements in terms of their probabilities. The following proposition shows that this is also reflected in the *semantics* by giving explicit examples of ProbFO-formulas that cannot be expressed in monodic ProbFO. Its proof relies on the main theorem in this chapter, Theorem 3.11

below, which implies that monodic ProbFO has the *finite world property*, that is, every satisfiable monodic ProbFO sentence is satisfiable in a model with only finitely many worlds.

Proposition 3.7. *The following ProbFO formulas are not expressible in monodic ProbFO:*

1. $w(P(x, y)) \sim p$ with P binary, $p \in (0, 1)$, and $\sim \in \{<, \leq, =, \geq, >\}$;
2. $w(A(x)) > w(A(y))$ with A unary.

Proof. For Point 1, assume that $\psi(x, y)$ is a formula expressing $w(P(x, y)) \sim p$. Now, consider the sentence $\varphi = \forall x (\varphi_1(x) \wedge \varphi_2(x) \wedge \varphi_3(x))$ with

$$\begin{aligned}\varphi_1(x) &= \neg A(x) \Rightarrow (w(A(x)) > 0); \\ \varphi_2(x) &= A(x) \Rightarrow \exists y (\psi(x, y) \wedge \neg A(y)); \\ \varphi_3(x) &= \neg A(x) \Rightarrow \forall y (\psi(x, y) \Rightarrow \neg A(y)),\end{aligned}$$

which was used before for showing the lack of the finite world property [101]. We repeat the argument for the sake of completeness. By Theorem 3.11, φ is satisfiable in a model with finitely many, say n , worlds. Fix some domain element d . By φ_1 , there are $k > 0$ worlds where d satisfies A . Fix one such world w . By formula φ_2 , there is a domain element d' not satisfying A in w that is related to d by $\psi(x, y)$. By formula φ_3 , in worlds w' where d did not satisfy A , also d' does not satisfy A . Thus, d' satisfies A in $k' < k$ many worlds. Continuing this argument leads to some domain element \bar{d} satisfying A in 0 worlds, in contradiction to φ_1 .

For Point 2, let $\psi(x, y)$ be a formula expressing $w(A(x)) > w(A(y))$ and assume a model $\mathfrak{M} = (D, W, \mu, \pi)$ of $\varphi = \forall x \exists y \psi(x, y)$. By Theorem 3.11, we can assume that W is finite. Fix some $d_0 \in D$ and an arbitrary $w \in W$. As \mathfrak{M} is a model of φ , there is an infinite sequence d_0, d_1, \dots such that $(\mathfrak{M}, w) \models \psi(d_i, d_{i+1})$ for each $i \geq 0$. Let p_i be the probability that d_i satisfies A . By the semantics of $\psi(x, y)$, we have $p_i > p_{i+1}$ for each $i \geq 0$. As W is finite, each p_i is a finite sum $\sum_{w \in W_i} \mu(w)$ for some $W_i \subseteq W$. However, this is a contradiction since there are only finitely many such subsets W_i . ■

To give concrete examples, formulas as under Point 1 of the Proposition could be used to express that any two persons who show up at a party together and both wear rings are probably married:

$$\forall xy (\text{wearsRing}(x) \wedge \text{wearsRing}(y) \wedge \text{comeTogether}(x, y)) \rightarrow w(\text{married}(x, y)) \geq 0.8.$$

With a formula as under Point 2., we could say that children are more likely to use a smartphone than their parents:

$$\forall xy \text{child}(x, y) \rightarrow w(\text{usesSmartphone}(x)) > 2 \cdot w(\text{usesSmartphone}(y)).$$

Reconsidering the street food example, we see that sentence (3.2) is not monodic as it makes a probabilistic statement about the binary predicate `tolerate`, and we can show similar to the proof of Proposition 3.7 that the same sentence is really not expressible in monodic ProbFO. The other formulas used in Example 3.2, however, are monodic. Intuitively, monodicity allows us to talk about the probabilities of single *individuals*. Thus, monodic ProbFO provides an object-centered view on degrees of belief, but features full expressivity on first-order level. Note that we can sometimes adapt non-monodic sentences to monodic ones expressing weaker statements. For instance, we can rewrite the first example above with

$$\forall xy \left(\text{wearsRing}(x) \wedge \text{wearsRing}(y) \wedge \text{comeTogether}(x, y) \right) \rightarrow \mathbf{w}(\exists y \text{ married}(x, y)) \geq 0.8$$

expressing that somebody wearing a ring who comes to a party with someone, who is also wearing a ring, is probably married.

3.2.2 Equality

We conclude this section by providing evidence that excluding equality is essential in the developments we present. The following theorem illustrates that recursive enumerability for monodic ProbFO relies on disallowing equality.

Theorem 3.8. *Validity in monodic ProbFO⁼ is not recursively enumerable.*

Proof. We reduce finite validity in FO. Let φ be an FO sentence, take a fresh unary predicate symbol P , and a fresh constant symbol c . Let ψ be the conjunction of the following formulas:

$$\begin{aligned} \psi_1 &= \mathbf{w}(P(c)) > 0 \\ \psi_2 &= \forall x \mathbf{w}(P(x)) \geq \mathbf{w}(P(c)) \\ \psi_3 &= \mathbf{w}(\forall x \forall y ((P(x) \wedge P(y)) \rightarrow x = y)) = 1 \end{aligned}$$

Let \mathfrak{M} be an arbitrary model of ψ . By ψ_1 , the constant c satisfies P with positive probability, say $r > 0$. By ψ_2 , every domain element satisfies P with at least r . Formula ψ_3 expresses that in every world, there is at most one object satisfying P , that is, the probabilities of all elements to satisfy P sum up to at most one. Thus, \mathfrak{M} can only have finitely many domain elements. Conversely, for every finite cardinality n , we can construct a model of ψ with domain size n . Hence, φ is finitely valid if, and only if, $(\psi \Rightarrow \varphi)$ is valid in ProbFO⁼. ■

This result is analogous to temporal first-order logic. Remarkably, there, Π_1^0 -hardness can be proved even when constant symbols are disallowed [131], whereas the proof of Theorem 3.8 crucially relies using them. Note that simulating the constant c in the proof of Theorem 3.8 using a fresh variable and equality results in loss of monodicity.

It remains open whether validity in monodic ProbFO⁼ without constant symbols is recursively enumerable. Here, we only show that there cannot be an analog to our main Theorem 3.11, which implies the finite world property.

Theorem 3.9. *Monodic ProbFO⁼ lacks the finite world property, even without constant symbols.*

Proof. Consider the conjunction of the following formulas:

$$\begin{aligned}\psi_1 &= \forall x \exists y R(x, y) \\ \psi_2 &= \forall xyz (R(x, y) \wedge R(y, z) \rightarrow R(x, z)) \\ \psi_3 &= \forall x \neg R(x, x) \\ \psi_4 &= \forall x w(A(x)) > 0 \\ \psi_5 &= w(\forall xy A(x) \wedge A(y) \rightarrow x = y) = 1\end{aligned}$$

Formulas ψ_1, ψ_2, ψ_3 enforce an infinite domain in a standard way, as they express totality, transitivity, and irreflexivity, respectively, of a binary relation R . It is known that such a relation can only be realized over an infinite domain. Formula ψ_4 ensures that for every such domain element, the probability of satisfying A is positive. However, by ψ_5 , there is only at most one element per world satisfying A . Thus, there have to be infinitely many worlds. ■

3.3 The Quasi-Model Machinery

We introduce quasi-models, an abstraction of probabilistic structures that underlies the proofs of all positive results established in this chapter. This requires some preliminary notation. For what follows, fix a monodic ProbFO-sentence φ_0 . We denote by:

- $\text{sub}(\varphi_0)$: the set of all subformulas of φ_0 and their negation;
- $\text{sub}_n(\varphi_0)$, $n \geq 0$: the formulas from $\text{sub}(\varphi_0)$ with precisely n free variables;
- $\text{con}(\varphi_0)$: the set of all constant symbols that occur in φ_0 .

Reflecting the monodicity condition, we mostly concentrate on subformulas with at most one free variable when defining quasi-models. Fix a fresh variable symbol x and define the set

$$\text{sub}_x(\varphi_0) = \text{sub}_0(\varphi_0) \cup \{\psi(x), \psi(c) \mid \psi(y) \in \text{sub}_1(\varphi_0), c \in \text{con}(\varphi_0)\}.$$

A *type* is a subset $t \subseteq \text{sub}_x(\varphi_0)$ such that \bar{t} is a maximal satisfiable subset of $\overline{\text{sub}_x(\varphi_0)}$.² Intuitively, a type is a set of FO formulas that are satisfied by a domain element in a world of some probabilistic structure; it also includes the FO formulas that are satisfied by the constants in that world. We say that two types t_1, t_2 *agree on sentences*, written $t_1 \equiv_0 t_2$, if for all sentences $\psi \in \text{sub}_x(\varphi_0)$, we have $\psi \in t_1$ iff $\psi \in t_2$. Clearly, only types that agree on sentences may be realized in the same world.

A *world type* is a set of types that agree on sentences; it can be viewed as an abstract representation of a world in a probabilistic structure, that is, of an FO structure. For an FO structure $\mathfrak{A} = (A, \pi)$ and an element $d \in A$, define

$$\begin{aligned} \text{tp}(\mathfrak{A}, d) &= \{\psi(x) \in \text{sub}_x(\varphi_0) \mid \mathfrak{A} \models \bar{\psi}[d]\}; \\ \text{tp}(\mathfrak{A}) &= \{\text{tp}(\mathfrak{A}, d) \mid d \in A\}. \end{aligned}$$

Note that $\text{tp}(\mathfrak{A}, d)$ is a type and $\text{tp}(\mathfrak{A})$ is a world type. A world type T is *realizable* if there is an FO structure \mathfrak{A} such that $\text{tp}(\mathfrak{A}) = T$, that is, if the FO formula $\bar{\chi}(T)$ is satisfiable, where $\chi(T)$ is defined as:

$$\chi(T) = \bigwedge_{t \in T} \exists x \bigwedge t(x) \wedge \forall x \bigvee_{t \in T} \bigwedge t(x).$$

Intuitively, $\chi(T)$ characterizes the world type T by stating that every type from T appears and no other types are realized.

Every probabilistic structure can be viewed as the set of world types that are realized. Thus, (collections of) world types will play a central role in the definition of quasi-models as they describe the first-order part. However, they need to be suitably enriched with

- (i) runs that describe the types of a single domain element in *all* worlds of a probabilistic structure; and
- (ii) relevant conditions that have to be satisfied by the probabilities of worlds.

For Point (i), let Q be a set of world types. A *run through* Q is a function r that assigns to each world type $T \in Q$ a non-empty set $r(T) \subseteq T$ and is *coherent*, that is, whenever some $t \in r(T)$ contains a weight formula θ , then for all $T' \in Q$ and $t' \in r(T')$, we have $\theta \in t'$. Coherence allows us to write $\theta \in r$ to denote that for all (equivalently: some) $T \in Q$ and $t \in r(T)$, we have $\theta \in t$. A run selects a set of types for each world type instead of only a single type because each world type can represent several actual worlds, and an element might have different types in each of these worlds. To finish Point (i), we combine world types and runs: A *quasi-model candidate* is a triple (T_0, Q, R) with T_0 a world type, Q a set of world types and R a set of runs through $Q \cup \{T_0\}$ such that for all

²Recall that $\bar{\psi}$ denotes the first-order formula that is obtained from ψ by replacing all weight formulas $\theta(\vec{x})$ by atoms $P_\theta(\vec{x})$; this is lifted to sets in the straightforward way.

$T \in Q \cup \{T_0\}$ and $t \in T$, there is a run $r \in R$ with $t \in r(T)$. Intuitively, T_0 describes a (single) world of probability 0 while each $T \in Q$ describes worlds of positive probability.

To address Point (ii) above and obtain our final quasi-model representation, we augment quasi-model candidates with a system of polynomial inequalities. It uses a variable x_T for each world type T to represent the probability of T (obtained by summing up the probabilities of all worlds of world type T) and a variable $x_{r,t,T}$ for each run r , world type T , and type $t \in T$ to describe the (summed up) probability of those worlds of world type T in which the element described by run r has type t .

Definition 3.10 (Quasi-Model). *A quasi-model candidate (T_0, Q, R) is a quasi-model for φ_0 if $\varphi_0 \in t$ for some $t \in T_0$, every $T \in Q \cup \{T_0\}$ is realizable, and the following system of polynomial inequalities $\mathcal{E}(Q, R)$ has a positive solution over the reals:*

1. *probability distribution on world types:*

$$\sum_{T \in Q} x_T = 1;$$

2. *the probabilities of the types associated by a run $r \in R$ to a world type $T \in Q$ sum up to the probability of T :*

$$x_T = \sum_{t \in r(T)} x_{r,t,T};$$

3. *runs respect weight formulas, that is, for all $f_1 \sim f_2 \in r$ with $\sim \in \{\leq, >\}$ ³ add the equation*

$$[f_1]_r \sim [f_2]_r,$$

where $[f]_r$ is obtained from f by replacing each outermost term $\mathbf{w}(\psi(x))$ with the following formula that describes probability of ψ :

$$\sum_{T \in Q} \sum_{t \in r(T), \psi(x) \in t} x_{r,t,T}.$$

Note that the field terms f_1, f_2 in Item 3 of Definition 3.10 can contain addition and multiplication, thus the system $\mathcal{E}(Q, R)$ need not be linear. We say that φ is *satisfied in a quasi-model* if there is a quasi-model for φ . The following provides the basis for our use of quasi-models in subsequent sections.

Theorem 3.11 (Main Theorem). *A monodic ProbFO sentence φ_0 is satisfiable iff it is satisfied in some quasi-model. Moreover, every satisfiable monodic ProbFO sentence is satisfied in a probabilistic structure with finitely many worlds.*

³We write $f_1 > f_2 \in r$ in place of $f_1 \leq f_2 \notin r$.

The remainder of this section is devoted to the proof of this theorem. In the “ \Rightarrow ” direction, we read off a quasi-model for φ_0 from a probabilistic structure that satisfies φ_0 . To show that the system $\mathcal{E}(Q, R)$ has a solution, the values for the variables x_T and $x_{r,t,T}$ are also read off in a straightforward way. More precisely, let $\mathfrak{M} = (D, W, \mu, \pi)$ be a probabilistic structure satisfying φ_0 , that is $(\mathfrak{M}, w_0) \models \varphi_0$. Observe first that we can w.l.o.g. assume that w_0 is the unique world with $\mu(w_0) = 0$: all worlds with probability 0 (except for w_0) can be dropped without changing \mathfrak{M} being a model of φ_0 . If after this transformation $\mu(w_0) > 0$ we can add a world w'_0 which is essentially a copy of w_0 with probability $\mu(w'_0) = 0$.

Now define a quasi-model (T_0, Q, R) for φ_0 . For this purpose, we lift the definition of the functions $\text{tp}(\cdot)$ to probabilistic structures:

$$\begin{aligned}\text{tp}(\mathfrak{M}, d, w) &= \{\psi \in \text{sub}_x(\varphi_0) \mid \mathfrak{M}, w \models \psi[d]\}; \\ \text{tp}(\mathfrak{M}, w) &= \{\text{tp}(\mathfrak{M}, d, w) \mid d \in D\}.\end{aligned}$$

Set $T_0 = \text{tp}(\mathfrak{M}, w_0)$ and $Q = \{\text{tp}(\mathfrak{M}, w) \mid w \in W, \mu(w) > 0\}$. Obviously, every $T \in Q \cup \{T_0\}$ is realizable. Next, define a set $R = \{r_d \mid d \in D\}$ where each function r_d is defined as

$$\begin{aligned}r_d(T_0) &= \{\text{tp}(\mathfrak{M}, d, w_0)\}; \\ r_d(T) &= \{\text{tp}(\mathfrak{M}, d, w) \mid w \in W, \mu(w) > 0, \text{tp}(\mathfrak{M}, w) = T\} \text{ for all } T \in Q.\end{aligned}$$

Obviously, each function r_d is a run through $\{T_0\} \cup Q$. To show that $\mathcal{E}(Q, R)$ is positively satisfiable, choose for each $r \in R$ a domain element $d(r) \in D$ such that $r_{d(r)} = r$. Then define values x_T^* for every $T \in Q$ and $x_{r,t,T}^*$ for every $r \in R$, $T \in Q$, and $t \in r(T)$ by taking:

$$\begin{aligned}x_T^* &= \mu(\chi(T)); \\ x_{r,t,T}^* &= \mu(\chi(T) \wedge t(d(r))).\end{aligned}$$

It remains to check that the values x_T^* , $x_{r,t,T}^*$ present a positive solution to $\mathcal{E}(Q, R)$ from Definition 3.10. Note first that all values are positive. For the remaining equations under Items 1 to 3, we have:

- The equations under Item 1 are satisfied as

$$\sum_{T \in Q} x_T^* = \sum_{T \in Q} \mu(\chi(T)) = \mu\left(\bigvee_{T \in Q} \chi(T)\right) = 1,$$

where the second equality holds as world types are pairwise contradictory, and the last equality holds since any world $w \in W$ satisfies $\chi(T)$ for some $T \in Q$.

- The equations under Item 2 are satisfied for every $r \in R$, $T \in Q$ since

$$\begin{aligned}
 \sum_{t \in r(T)} x_{r,t,T}^* &= \sum_{t \in r(T)} \mu(\chi(T) \wedge t(d(r))) \\
 &= \sum_{t \in T} \mu(\chi(T) \wedge t(d(r))) \\
 &= \mu(\chi(T) \wedge \bigvee_{t \in T} t(d(r))) \\
 &= \mu(\chi(T)) = x_T^*.
 \end{aligned}$$

The second equality holds since $\mu(\chi(T) \wedge t(d(r))) = 0$ in case $t \notin r(T)$. The third equality holds as types are pairwise contradictory. The fourth equality holds as $\chi(T) \Rightarrow \bigvee_{t \in T} t(x)$ is a tautology.

- For checking the equations under Item 3, let $f_1 \sim f_2 \in r$ for some $r \in R$. It suffices to show that $[f_i]_r = [f_i(d(r))]_{\mathfrak{M},w}$ for all $w \in W$ and $i \in \{1, 2\}$. Note that each f_i has at most one free variable, whose valuation we indicate by substituting $d(r)$ into the term f_i , then dispensing with further mention of a valuation. We make an induction on the structure of field terms f . The cases when f equals 0, 1, $f' + f''$, or $f' \times f''$ are clear. So it remains to consider the case $f = \mathbf{w}(\psi(x))$.

$$\begin{aligned}
 [\mathbf{w}(\psi(d(r)))]_{\mathfrak{M},w} &= \mu(\psi(d(r))) \\
 &= \mu\left(\bigvee_{t \in T, \psi(x) \in t} (\chi(T) \wedge t(d(r)))\right) \\
 &= \sum_{T \in Q} \sum_{t \in T, \psi(x) \in t} \mu(\chi(T) \wedge t(d(r))) \\
 &= \sum_{T \in Q} \sum_{t \in r(T), \psi(x) \in t} \mu(\chi(T) \wedge t(d(r))) \\
 &= \sum_{T \in Q} \sum_{t \in r(T), \psi(x) \in t} x_{r,t,T}^*.
 \end{aligned}$$

The first equality is just the semantics. The second equality holds as the big disjunction covers all possibilities. The third equality holds as both distinct world types and distinct types are contradictory. The last equality holds as $\mu(\chi(T) \wedge t(d(r))) = 0$ in case $t \notin r(T)$.

It remains to remark that by assumption $\varphi_0 \in t$ for all $t \in T_0$ and thus (T_0, Q, R) is a quasi-model for φ_0 .

For the “ \Leftarrow ”-direction, let (T_0, Q, R) be a quasi-model for φ_0 . Hence, every $T \in Q \cup \{T_0\}$ is realizable and $\mathcal{E}(Q, R)$ has a positive solution which we denote by x_T^* for $T \in Q$ and

$x_{r,t,T}^*$ for $r \in R$, $T \in Q$, $t \in r(T)$. To construct a probabilistic structure \mathfrak{M} that satisfies φ_0 , we *cannot* use the world types in $Q \cup \{T_0\}$ directly as worlds, since runs can associate more than one type with a world type. We thus need to subdivide each $T \in Q$ into several worlds, each accommodating a *single* type that a given run assigns to T . This has to be done in a careful way since we have to do it simultaneously for *all runs* while also ensuring that all types in T are realized in each of the worlds that T is subdivided into. Note that world type T_0 is an exception to what was said above, since we can assume w.l.o.g. that $r(T_0)$ is a singleton set for each $r \in R$: As the types in T_0 do not contribute to the equation system $\mathcal{E}(Q, R)$, we can replace a run r with $r(T_0) = \{t_1, \dots, t_k\}$ by k runs r_1, \dots, r_k defined as $r_i(T_0) = \{t_i\}$ and $r_i(T) = r(T)$ for all $T \in Q$ and $i \in \{1, \dots, k\}$. For the remaining world types from Q we need the following notion.

Definition 3.12 (Subdivision). *Let $r \in R$ and $T \in Q$. A subdivision of T for r is a tuple $s = (b_1, \dots, b_n, \zeta)$ such that $0 < b_1 < b_2 < \dots < b_n = x_T^*$, $n = |r(T)| + 1$, and ζ is a surjective function that assigns to every b_i a type $\zeta(b_i) \in r(T)$ such that for all $t \in r(T)$ we have $\sum_{i \in [1, n], \zeta(b_i) = t} (b_i - b_{i-1}) = x_{r,t,T}^*$, where, here and in what follows, $b_0 := 0$.*

Intuitively, the interval $[0, x_T^*]$ represents the probability covered by all worlds of type T and we subdivide this range into the intervals $(b_i, b_{i+1}]$, with $i < n$. Elements described by the run r then have type $\zeta(b_{i+1})$ in the interval $(b_i, b_{i+1}]$. For easier reference, we say for $p \in (0, x_T^*]$ that s has type t at p if $p \in (b_{i-1}, b_i]$ and $\zeta(b_i) = t$. We next lift this to runs.

A *subdivided run* is a pair (r, S) with $r \in R$ a run and S a function that associates with every $T \in Q$ a subdivision $S(T)$ of T for r . We could use the subintervals identified by a subdivided run (r, S) as worlds if we had only the single run r . Since this is not the case, though, worlds are defined by combining a sufficiently rich set of subdivided runs in an appropriate way. This set is identified by the next claim.

Claim 1. There is a finite set Γ of subdivided runs such that

- (*) for all $r \in R, T \in Q, t \in r(T)$, and $p \in (0, x_T^*]$, there is some $(r, S) \in \Gamma$ such that $S(T)$ has type t at p .

Proof of Claim 1. It suffices to show the statement for an arbitrary fixed $r \in R$, $T \in Q$, $t \in T$. By assumption, $x_{r,t,T}^* > 0$. Obviously, there is a finite set of intervals $(y_1, z_1], \dots, (y_k, z_k]$ each of length $x_{r,t,T}^*$ which cover $(0, x_T^*]$. For each such interval $(y_i, z_i]$ we can find a subdivision $s_i = (b_1, \dots, b_n, \zeta)$ of T for r such that s_i has type t at p for all $p \in (y_i, z_i]$: since $n > r(T)$, we can always arrange the n intervals of a subdivision such that there is a j with $(b_{j-1}, b_j] = (y_i, z_i]$ and $\zeta(b_j) = t$, see Figure 3.2. We have thus fixed k subdivisions s_1, \dots, s_k of T for r . Moreover, note that, for $T' \in Q$, $T' \neq T$, we can trivially fix a subdivision $s_{T'}$ of T' for r . Combining those with the subdivisions

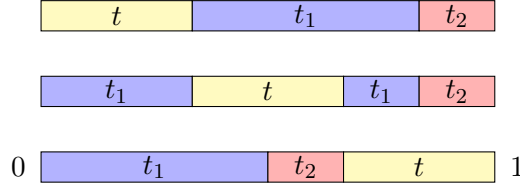


Figure 3.2: Idea underlying the Proof of Claim 1. Assume $r(T) = \{t, t_1, t_2\}$ with $x_{r,t,T}^* = 1/3$, $x_{r,t_1,T}^* = 1/2$, and $x_{r,t_2,T}^* = 1/6$. The types t, t_1, t_2 are represented by bars of colors yellow, blue, and red, respectively. Each row represents one subdivision of the types; together the subdivisions ensure that for each $p \in [0, 1]$ there is a row where t is satisfied at p . The middle row demonstrates why we need to divide in at most 4 pieces.

obtained above yields k subdivided runs $(r, S_1), \dots, (r, S_k)$ where

$$S_i(T') = \begin{cases} s_i & \text{if } T' = T \\ s_{T'}; & \text{otherwise.} \end{cases}$$

This finishes the proof of Claim 1.

We are now ready to define a probabilistic structure $\mathfrak{M} = (D, W, \mu, \pi)$ that satisfies φ_0 . We start with the set of worlds W and the distribution μ . Let Γ be the set of subdivided runs from Claim 1. For every $T \in Q$, let $\mathfrak{Z}(T)$ denote the set of all values b_i that occur in a subdivision $s = (b_1, \dots, b_n, \zeta)$ for T and r , for some $(r, S) \in \Gamma$ with $S(T) = s$. Further assume that $\mathfrak{Z}(T) = \{z_1^T, \dots, z_{m_T}^T\}$ with $z_1^T < \dots < z_{m_T}^T$. Now set

$$\begin{aligned} W &= \{(T, z) \mid T \in Q, z \in \mathfrak{Z}(T)\} \cup \{(T_0, 0)\}; \\ \mu(T, z_i^T) &= z_i^T - z_{i-1}^T \quad \text{for all } T \in Q \text{ and } 1 \leq i \leq m_T \text{ (where } z_0^T := 0); \\ \mu(T_0, 0) &= 0. \end{aligned}$$

Notice that every world (T, z) with $T \in Q$ has positive probability and the additional world $(T_0, 0)$ has probability 0.

Next, we define the domain D of \mathfrak{M} . By assumption, every $T \in Q \cup \{T_0\}$ is realizable. Hence, for every world type $T \in Q \cup \{T_0\}$ we can fix an FO structure \mathfrak{A}_T that realizes it. Let $\mathfrak{A}_{T,z} = (A_{T,z}, \pi_{T,z})$, for $(T, z) \in W$, be pairwise disjoint copies of \mathfrak{A}_T and define D to be the disjoint union of all $A_{T,z}$, $(T, z) \in W$.

It remains to give the interpretation function π of \mathfrak{M} . For this purpose, we assign to every $d \in D$ a function σ_d that associates every world with a type. Given d , we first determine the unique T, z such that $d \in A_{T,z}$ and choose a subdivided run as follows:

- If $T \in Q$, then choose some $(r, S) \in \Gamma$ that has type $\text{tp}(\mathfrak{A}_{T,z}, d)$ at z (possible due to Claim 1);

- if $T = T_0$, choose some $(r, S) \in \Gamma$ with $r(T_0) = \{\text{tp}(\mathfrak{A}_{T_0,0}, d)\}$.

For all $(T', z') \in W$ with $T' \neq T_0$, define $\sigma_d(T', z')$ as the type of $S(T')$ at z' ; additionally, set $\sigma_d(T_0, 0) = t_0$, where $r(T_0) = \{t_0\}$.

Let us verify that in every world $(T, z) \in W$ precisely the types from T appear in the following sense:

- (i) for all $t \in T$, there is some d such that $\sigma_d(T, z) = t$;
- (ii) if $\sigma_d(T, z) = t$, then $t \in T$.

For Point (i), observe that by construction there is some $d \in A_{T,z}$ realizing t . By the choice of the run used to define σ_d , we get $\sigma_d(T, z) = t$. Point (ii) is trivial in case $T = T_0$. For $T \neq T_0$, observe that $\sigma_d(T, z)$ is always the type of some subdivision of T for some run r , which is always a type from T .

We next show that the types given by the mappings σ_d are realizable in the sense that, for each $(T, z) \in W$, there is a model $\mathfrak{B}_{T,z} = (D, \pi'_{T,z})$ such that for all $d \in D$ we have:

$$\sigma_d(T, z) = t \quad \text{iff} \quad \text{tp}(\mathfrak{B}_{T,z}, d).$$

In fact, the existence of such a model is an immediate consequence of the definition of σ_d and the following standard result from model theory which can be proved based on the fact that we consider FO *without* equality. Intuitively, we can obtain $\mathfrak{B}_{T,z}$ from $\mathfrak{A}_{T,z}$ by duplicating domain elements.

Lemma 3.13. *If a world type T is realizable in a structure $\mathfrak{A} = (A, \pi)$ such that every type $t \in T$ is realized by κ_t elements in A , then for any family of cardinals λ_t , $t \in T$, with $\lambda_t \geq \kappa_t$ for all $t \in T$, there is a structure $\mathfrak{A}' = (A', \pi')$ such that every type $t \in T$ is realized by precisely λ_t elements in A' .*

To finish the definition of \mathfrak{M} , set for all predicate names P and all worlds $(T, z) \in W$:

$$\pi(P, (T, z)) = \pi'_{T,z}(P).$$

In order to prove correctness of the construction we show the following claim. There, it is convenient to write $\psi(x)$ even if ψ is a sentence, i.e., x does not occur freely in ψ .

Claim 2. For all valuations ν , all $\psi(x) \in \text{sub}_x(\varphi_0)$, and all $(T, z) \in W$ we have

$$(\mathfrak{B}_{T,z}, \nu) \models \overline{\psi(x)} \quad \Longleftrightarrow \quad (\mathfrak{M}, (T, z), \nu) \models \psi(x).$$

Proof of Claim 2. The induction base, i.e., $\psi(x) = P(x_1, \dots, x_k)$, is clear since then $\psi(x) = \overline{\psi(x)}$ and, by construction, $\pi'_{T,z}(P) = \pi(P, (T, z))$. The induction steps for the constructors \wedge , \neg , $\forall x$ follow immediately from induction hypothesis.

Now let $\psi(x) = f_1 \leq f_2$ and thus $\overline{\psi(x)} = P_\psi(x)$. Assume $\nu(x) = d$ and $d \in A_{T^*, z^*}$ for some $(T^*, z^*) \in W$, and fix the subdivided run (r, S) used in the construction of σ_d . By definition of π , $(\mathfrak{B}_{T, z}, \nu) \models P_\psi(x)$ is equivalent to $\psi(x) \in \text{tp}(\mathfrak{B}_{T, z}, d)$. As $\psi(x)$ is a weight formula and r satisfies the coherence condition, this is equivalent to the fact that $\psi(x) \in r$. By Item 3 of Definition 3.10 and maximality of types, this is the case if, and only if, it holds $[f_1]_r \leq [f_2]_r$ (with the values $x_{r, t, T}^*$ from the fixed solution). Now observe that subdivisions and subdivided runs do not change the probability of some type t , but only give some arrangement. Thus, by construction of the set of worlds, $[f_1]_r \leq [f_2]_r$ is equivalent to $[f_1]_{(\mathfrak{M}, w, \nu)} \leq [f_2]_{(\mathfrak{M}, w, \nu)}$ for any $w \in W$. Finally, this is equivalent to $(\mathfrak{M}, (T, z), \nu) \models f_1 \leq f_2$. This finishes the proof of Claim 2.

We finally verify that φ_0 is satisfied in some world of \mathfrak{M} ; in particular, we show that $(\mathfrak{M}, (T_0, 0)) \models \varphi_0$. By definition of a quasi-model, we have $\varphi_0 \in t$ for some $t \in T_0$. By construction of \mathfrak{M} , there is some $d \in D$ with $\mathfrak{B}_{T_0, 0} \models \bar{t}(d)$. As $\varphi_0 \in t$ is a sentence, $(\mathfrak{B}_{T_0, 0}, \nu) \models \overline{\varphi_0}$ for any valuation ν . By the above claim, $(\mathfrak{M}, (T_0, 0), \nu) \models \varphi_0$ for any valuation ν , and thus $(\mathfrak{M}, (T_0, 0)) \models \varphi_0$.

It remains to note that Claim 1 and the definition of the set of worlds W in \mathfrak{M} imply that \mathfrak{M} is a model comprising only finitely many worlds. Thus, every satisfiable monodic ProbFO sentence is satisfied in a model with finitely many worlds. This finishes the proof of Theorem 3.11. \blacksquare

Recall that our semantics interprets constants in a non-rigid way, that is, the interpretation of constants may differ in different worlds. However, quasi-models and all results stated below can be adapted to rigid constants. A world type is then a pair $\langle T, \{t_c \mid c \in \text{con}(\varphi_0)\} \rangle$ with T a set of types and $t_c \in T$. We call $\langle T, \{t_c \mid c \in \text{con}(\varphi_0)\} \rangle$ *realizable* if there is an FO structure $\mathfrak{A} = (A, \pi)$ such that $\text{tp}(\mathfrak{A}) = T$ and $\text{tp}(\mathfrak{A}, \pi(c)) = t_c$ for each $c \in \text{con}(\varphi_0)$. For a quasi-model candidate (T_0, Q, R) , we additionally require R to contain, for each $c \in \text{con}(\varphi_0)$, a run r_c defined by $r_c(T) = \{t_c\}$ for each $\langle T, \{t_{c'} \mid c' \in \text{con}(\varphi_0)\} \rangle \in Q$. It is now easy to adapt the proof of Theorem 3.11.

3.4 Recursive Enumerability and Axiomatization

The first application of Theorem 3.11 is to show that validity in monodic ProbFO is recursively enumerable and hence, our definition of monodicity is suitable. Moreover, we provide a concrete axiomatization.

For the former, it suffices to devise a semi-decision procedure for unsatisfiability. The crucial observation is that, for any input sentence φ_0 , the number of quasi-model candidates (T_0, Q, R) is bounded. It is thus possible to construct all quasi-model candidates (T_0, Q, R) such that φ_0 is contained in some $t \in T_0$ and then eliminate those for which the system of polynomial inequalities $\mathcal{E}(Q, R)$ from Definition 3.10 is not satisfiable. Then, enumerate all unsatisfiable FO formulas. For each such formula ψ , eliminate all

<i>PC</i>	an axiomatization of FO;
<i>OF</i>	all instances of the axioms of ordered fields that are well-formed formulas in monodic ProbFO;
<i>PW₁</i>	$\varphi \Rightarrow (\mathbf{w}(\varphi) = 1)$, if all occurrences of predicate symbols in φ are inside the scope of some $\mathbf{w}()$;
<i>PW₂</i>	$\mathbf{w}(\varphi) \geq 0$;
<i>PW₃</i>	$\mathbf{w}(\varphi \wedge \psi) + \mathbf{w}(\varphi \wedge \neg\psi) = \mathbf{w}(\varphi)$;
<i>PW₄</i>	$\mathbf{w}(\exists x \varphi(x)) > 0 \Rightarrow \exists x \mathbf{w}(\varphi(x)) > 0$;
<i>RPW</i>	from $\varphi \equiv \psi$ infer $\mathbf{w}(\varphi) = \mathbf{w}(\psi)$.

Figure 3.3: Axiomatization for monodic ProbFO.

quasi-model candidates (T_0, Q, R) such that $\bar{\chi}(T) = \psi$ for some $T \in Q \cup \{T_0\}$: unsatisfiability of $\bar{\chi}(T)$ implies that T is not realizable, thus (T_0, Q, R) cannot be a quasi-model. Once all quasi-model candidates have been eliminated, return with ‘ φ_0 is unsatisfiable’.

Theorem 3.14. *The set of valid monodic ProbFO sentences is recursively enumerable.*

Although we have already shown recursive enumerability, axiomatizations are of independent theoretical interest. Halpern gives an axiomatization of ProbFO for the case where probabilistic structures are restricted to a domain of bounded size [67]. We propose a variation of this axiomatization that is sound and complete for monodic ProbFO, without assuming bounded domains.

Let AX be the set of axioms in Figure 3.3. In comparison to Halpern’s axiomatization, we have removed the axiom FIN_N for bounded domains of size N and added axiom PW_4 . This axiom follows from Halpern’s axiomatization, but is independent of the axioms that remain when FIN_N is removed – in a nutshell, its soundness over discrete measures depends on σ -additivity, while PW_3 captures only finite additivity.⁴ Moreover, as we exclude field variables, we no longer need the full axiomatization of real-closed fields, but can make do with the axioms of ordered fields, which are the field axioms together with axioms describing a total order \leq compatible with the field operations, see Figure 3.4. These axioms are instantiated to monodic ProbFO formulas by replacing real variables with weight terms, observing the monodicity restriction. As an example, from the commutativity axiom for $+$, we get the axiom $\forall x \mathbf{w}(\varphi(x)) + \mathbf{w}(\psi(x)) = \mathbf{w}(\psi(x)) + \mathbf{w}(\varphi(x))$ for all monodic ProbFO formulas $\varphi(x), \psi(x)$ with at most one free variable x . Thus, *OF*

⁴Thanks to Lutz Schröder for pointing this out.

commutativity $\forall x \forall y (x \circ y) = (y \circ x)$ for $\circ \in \{+, \times\}$;

associativity $\forall x \forall y \forall z (x \circ (y \circ z)) = ((x \circ y) \circ z)$ for $\circ \in \{+, \times\}$;

identity $\forall x (x + 0) = x$ and $\forall x (x \times 1) = x$;

inverse $\forall x \exists \bar{x} (x + \bar{x}) = 0$ and $\forall x (x \neq 0) \rightarrow \exists \bar{x} (x \times \bar{x}) = 1$;

distributivity $\forall x \forall y \forall z (x \times (y + z)) = ((x \times y) + (x \times z))$;

total order $\forall x \forall y (x \leq y) \vee (y \leq x)$, $\forall x \forall y (x \leq y) \wedge (y \leq x) \Rightarrow (x = y)$, and
 $\forall x \forall y \forall z (x \leq y) \wedge (y \leq z) \Rightarrow (x \leq z)$;

compatibility($\leq, +$) $\forall x \forall y \forall z (x \leq y) \Rightarrow (x + z \leq y + z)$;

compatibility(\leq, \times) $\forall x \forall y (x \geq 0) \wedge (y \geq 0) \Rightarrow (x \times y \geq 0)$.

Figure 3.4: Axiomatization for ordered fields.

enables us to apply standard arithmetical laws to weight formulas. Finally, note that—as observed in [67]—there is a slight adaptation necessary for the FO axiom $\forall x \varphi \Rightarrow \varphi[x/t]$ *whenever t is substitutable for x in φ* , which is intuitively the case when t does not contain a variable y that ends up bounded after this substitution. More precisely, as constants are non-rigid designators, the instance $\forall x \mathbf{w}(A(x)) = 1/2 \Rightarrow \mathbf{w}(A(a)) = 1/2$ of this axiom is not valid. Thus, we need to prohibit the substitution of constant symbols into weight formulas in order to retain soundness of the axiom.

Theorem 3.15. *AX axiomatizes validity in monodic ProbFO.*

For the axioms except for PW_4 , soundness is proved essentially as in [67]. For the additional axiom PW_4 assume some probabilistic structure $\mathfrak{M} = (D, W, \pi, \mu)$ and some valuation ν such that $(\mathfrak{M}, w, \nu) \models \mathbf{w}(\exists x \varphi(x)) > 0$ for some world $w \in W$. We have:

$$\begin{aligned}
 (\mathfrak{M}, w, \nu) \models \mathbf{w}(\exists x \varphi(x)) > 0 &\Rightarrow \exists w' \in W : \mu(w') > 0 \text{ and } (\mathfrak{M}, w', \nu) \models \exists x \varphi(x) \\
 &\Rightarrow \exists d \in D : (\mathfrak{M}, w', \nu[x \rightarrow d]) \models \varphi(x) \\
 &\Rightarrow (\mathfrak{M}, w, \nu[x \rightarrow d]) \models \mathbf{w}(\varphi(x)) > 0 \\
 &\Rightarrow (\mathfrak{M}, w, \nu) \models \exists x \mathbf{w}(\varphi(x)) > 0.
 \end{aligned}$$

For showing completeness, we need some more notation. We use $AX \vdash \varphi$ to denote that φ can be derived in AX and call a sentence φ *consistent* if $AX \not\vdash \neg\varphi$. By Theorem 3.11, it suffices the following lemma.

Lemma 3.16. *If a monodic ProbFO sentence φ_0 is consistent, then there is a quasi-model for φ_0 .*

We first prove some auxiliary statements. For the sake of simplicity, we will sometimes use $PC \vdash \varphi$ (instead of $PC \vdash \bar{\varphi}$) to express that $\bar{\varphi}$ can be derived using the axioms from PC . Note that this does not lead to confusion as $PC \vdash \bar{\varphi}$ implies $AX \vdash \varphi$. We say that $\varphi_1, \dots, \varphi_m$ are *pairwise mutually exclusive* if $PC \vdash \varphi_i \Rightarrow \neg \varphi_j$ for $i \neq j$.

Lemma 3.17. (1) *If $\varphi_1, \dots, \varphi_k$ are pairwise mutually exclusive, then $AX \vdash \mathbf{w}(\varphi_1 \vee \dots \vee \varphi_k) = \mathbf{w}(\varphi_1) + \dots + \mathbf{w}(\varphi_k)$.*

(2) *If $AX \vdash \varphi$, then $AX \vdash \mathbf{w}(\varphi) = 1$.*

(3) *$AX \vdash \mathbf{w}(\varphi) + \mathbf{w}(\neg \varphi) = 1$.*

(4) *$AX \vdash \mathbf{w}(\varphi \wedge \psi) \leq \mathbf{w}(\varphi)$.*

(5) *$AX \vdash \mathbf{w}(\varphi \wedge \theta) > 0 \Rightarrow \theta$ provided that every predicate symbol in θ appears inside a weight term $\mathbf{w}(\psi)$.*

(6) *$AX \vdash \sum_{\psi \in \Psi} \mathbf{w}(\psi) = r \Rightarrow \bigvee_{\Psi' \subseteq \Psi} \left(\bigwedge_{\psi \in \Psi'} (\mathbf{w}(\psi) > 0) \wedge \sum_{\psi \in \Psi'} \mathbf{w}(\psi) = r \right)$.*

Proof. Items (1)–(4) are shown in [67]. For Item (5), assume the contrary, that is, $\mathbf{w}(\varphi \wedge \theta) > 0 \wedge \neg \theta$ is consistent. By Axiom PW_1 , also $\mathbf{w}(\varphi \wedge \theta) > 0 \wedge \mathbf{w}(\neg \theta) = 1$ is consistent. By Item (3) of this Lemma and OF , also $\mathbf{w}(\varphi \wedge \theta) > 0 \wedge \mathbf{w}(\theta) = 0$ is consistent. By Item (4), we get that $\mathbf{w}(\theta) > 0 \wedge \mathbf{w}(\theta) = 0$ is consistent, which is in contradiction to OF .

For Item (6), observe that, by axiom PW_2 , we have $AX \vdash \sum_{\psi \in \Psi} \mathbf{w}(\psi) = r \Rightarrow \sum_{\psi \in \Psi} \mathbf{w}(\psi) = r \wedge \bigwedge_{\psi \in \Psi} \mathbf{w}(\psi) \geq 0$. By standard arithmetics, $\mathbf{w}(\psi) \geq 0$ is equivalent to $(\mathbf{w}(\psi) = 0) \vee (\mathbf{w}(\psi) > 0)$. This, and distributivity of \vee over \wedge now leads to

$$AX \vdash \sum_{\psi \in \Psi} \mathbf{w}(\psi) = r \Rightarrow \bigvee_{\Psi' \subseteq \Psi} \left(\bigwedge_{\psi \in \Psi'} (\mathbf{w}(\psi) > 0) \wedge \bigwedge_{\psi \in \Psi \setminus \Psi'} (\mathbf{w}(\psi) = 0) \wedge \sum_{\psi \in \Psi} \mathbf{w}(\psi) = r \right).$$

Since 0 is neutral with respect to addition, we can just omit those ψ such that $\mathbf{w}(\psi) = 0$ in the inner sum and obtain the desired derivation. \blacksquare

We are ready to prove Lemma 3.16. For a type t , let us denote with $\mathbf{wf}(t)$ the set of all weight formulas or their negations contained in t . Moreover, we say that two types t, t' *agree on weight formulas* if $\mathbf{wf}(t) = \mathbf{wf}(t')$. Denote the set of realizable world types with W . Note that we can concentrate on realizable world types, since for each unrealizable world type T , we have $PC \vdash \neg \chi(T)$. We assume that φ_0 is consistent and begin with showing the following claim.

Claim 1. There is a realizable world type T_0 , a type $t \in T_0$ with $\varphi_0 \in t$, and a set Q of realizable world types such that

(i) the following formula is consistent:

$$\vartheta = \chi(T_0) \wedge \left(\sum_{T \in Q} \mathbf{w}(\chi(T)) = 1 \right) \wedge \left(\bigwedge_{T \in Q} \mathbf{w}(\chi(T)) > 0 \right);$$

(ii) for each $T \in Q$, $t \in T$ there is a $t_0 \in T_0$ agreeing on the weight formulas with t ;

(iii) for each $T \in Q$, $t_0 \in T_0$ there is a $t \in T$ agreeing on the weight formulas with t_0 .

Proof of Claim 1. We begin by noting that $PC \vdash \varphi_0 \equiv \bigvee_{T \in W_0} \chi(T)$, where W_0 is the set of all realizable world candidates containing a type t with $\varphi_0 \in t$. Thus, consistency of φ_0 implies that $\chi(T_0)$ is consistent for some realizable world type T_0 containing some $t \in T_0$ with $\varphi_0 \in t$. Next, observe that $PC \vdash \bigvee_{T \in W} \chi(T)$, and thus, by Lemma 3.17(2), $AX \vdash \mathbf{w}(\bigvee_{T \in W} \chi(T)) = 1$. As for distinct world types T, T' we know that $\chi(T)$ and $\chi(T')$ are mutually exclusive, Lemma 3.17(1) implies $AX \vdash \sum_{T \in W} \mathbf{w}(\chi(T)) = 1$. As $\chi(T_0)$ is consistent, also the formula $\chi(T_0) \wedge \sum_{T \in W} \mathbf{w}(\chi(T)) = 1$ is consistent. By Lemma 3.17(6), we can identify a subset $Q \subseteq W$ such that ϑ is consistent; thus, we are finished with Item (i).

For Item (ii), assume some $T \in Q$ and a type $t \in T$ that does not agree on the weight formulas with any type from T_0 . Observe that $AX \vdash \chi(T_0) \Rightarrow \forall x \bigvee_{t_0 \in T_0} \mathbf{wf}(t_0(x))$. On the other hand, we have that $AX \vdash \vartheta \Rightarrow \mathbf{w}(\chi(T)) > 0$ and thus $AX \vdash \vartheta \Rightarrow \mathbf{w}(\exists x t(x)) > 0$ by Lemma 3.17(4). By PW_4 , we obtain $AX \vdash \vartheta \Rightarrow \exists x \mathbf{w}(t(x)) > 0$. Applying Lemma 3.17(5) yields $AX \vdash \vartheta \Rightarrow \exists x \mathbf{wf}(t(x))$. Overall, consistency of ϑ yields consistency of $\forall x (\bigvee_{t_0 \in T_0} \mathbf{wf}(t_0(x))) \wedge \exists x \mathbf{wf}(t(x))$ which is a contradiction by PC , the assumption that t does not agree on the weight formulas with any $t_0 \in T$, and maximality of types. Item (iii) can be proved analogously. This finishes the proof of Claim 1.

The world type T_0 and the set of world types Q identified in Claim 1 establish the basis for the quasi-model whose existence we are going to prove. In order to show how to use consistency of ϑ to also identify a suitable set of runs, we need some auxiliary formulas that can be derived in AX .

Claim 2. For every formula $\psi \in \mathbf{sub}_x(\varphi_0)$ and all $S \in W$, the following can be derived:

$$AX \vdash \vartheta \Rightarrow \mathbf{w}(\psi(x)) = \sum_{S \in Q} \sum_{s \in S, \psi(x) \in s} \mathbf{w}(\chi(S) \wedge s(x)); \quad (3.18)$$

$$AX \vdash \mathbf{w}(\chi(S)) = \sum_{s \in S} \mathbf{w}(\chi(S) \wedge s(x)). \quad (3.19)$$

Proof of Claim 2. We start with deriving (3.18). First observe that we have $PC \vdash \psi(x) \equiv \bigvee_{S \in \widehat{W}} \bigvee_{s \in S, \psi(x) \in s} (\chi(S) \wedge s(x))$; thus RPW and Lemma 3.17(1) lead to

$$AX \vdash \mathbf{w}(\psi(x)) = \sum_{S \subseteq \widehat{W}} \sum_{s \in S, \psi(x) \in s} \mathbf{w}(\chi(S) \wedge s(x)).$$

It remains to note that that $AX \vdash \vartheta \Rightarrow \mathbf{w}(\chi(S)) = 0$ for all $S \notin Q$; by Lemma 3.17(4), also $AX \vdash \vartheta \Rightarrow \mathbf{w}(\chi(S) \wedge s(x)) = 0$ for all $s \in S$.

For deriving (3.19), observe that we have for all world types $S \in W$:

$$PC \vdash \chi(S) \equiv \chi(S) \wedge \bigvee_{s \in S} s(x).$$

Hence, $AX \vdash \mathbf{w}(\chi(S)) = \mathbf{w}(\chi(S) \wedge \bigvee_{s \in S} s(x))$, and Lemma 3.17(1) leads to the result. This finishes the proof of Claim 2.

The next step is to show how consistency of ϑ induces a run through every type $t \in T$, $T \in Q$. For this purpose, fix some $T \in Q$, a type $t \in T$, and the set $\mathbf{wf}(t) = \{\theta_1, \dots, \theta_m\}$ of weight formulas (or their negations) in t . Obviously, we have $AX \vdash \vartheta \Rightarrow \mathbf{w}(\chi(T)) > 0$. As $\chi(T)$ contains the conjunct $\exists x t(x)$, we have $AX \vdash \vartheta \Rightarrow \exists x \mathbf{w}(\chi(T) \wedge t(x)) > 0$ using PW_4 . By applying Lemma 3.17(5) to $\theta_1, \dots, \theta_m$, we can also derive

$$AX \vdash \vartheta \Rightarrow \exists x \mathbf{w}(\chi(T) \wedge t(x)) > 0 \wedge \theta_1(x) \wedge \dots \wedge \theta_m(x) \quad (3.20)$$

By (3.18) and standard arithmetical laws (consequences of OF), we can replace each $\theta_j(x)$ in (3.20) by the formula $\theta'_j(x)$ that is obtained by substituting every $\mathbf{w}(\psi(x))$ with $\sum_{S \in Q} \sum_{s \in S, \psi(x) \in s} \mathbf{w}(\chi(S) \wedge s(x))$. Moreover, we can add formula (3.19) for every $S \in Q$ as conjunct. We thus obtain the following:

$$AX \vdash \vartheta \Rightarrow \exists x \left(\mathbf{w}(\chi(T) \wedge t(x)) > 0 \wedge \theta'_1(x) \wedge \dots \wedge \theta'_m(x) \wedge \bigwedge_{S \in Q} \mathbf{w}(\chi(S)) = \sum_{s \in S} \mathbf{w}(\chi(S) \wedge s(x)) \right)$$

Recall that $AX \vdash \vartheta \Rightarrow \mathbf{w}(\chi(S)) > 0$ for each $S \in Q$. Thus, in the sum in the second line, $\mathbf{w}(\chi(S) \wedge s(x))$ has to be positive for at least one $s \in S$. Using Lemma 3.17(6), one can show that there exists a choice function selecting the set of all such s for each S . More precisely, there is a function r such that $t \in r(T)$ (due to the conjunct $\mathbf{w}(\chi(T) \wedge t(x)) > 0$), $r(S) \neq \emptyset$ for all $S \in Q$, and the following is consistent:

$$\begin{aligned} \vartheta \wedge \exists x \left(\bigwedge_{j=1}^m \theta'_j(x) \wedge \bigwedge_{S \in Q, s \in r(S)} \mathbf{w}(\chi(S) \wedge s(x)) > 0 \right. \\ \left. \bigwedge_{S \in Q} \mathbf{w}(\chi(S)) = \sum_{s \in r(S)} \mathbf{w}(\chi(S) \wedge s(x)) \right) \end{aligned}$$

Observe now that all types $s \in r(S)$, $S \in Q$ actually agree on the weight formulas; in particular, for all such s we have that $\mathbf{wf}(s) = \mathbf{wf}(t)$. Assume to the contrary, $\neg\theta_j(x) \in s$ for some $s \in r(S)$. By Lemma 3.17(5), we have $AX \vdash \mathbf{w}(\psi(S) \wedge s(x)) > 0 \Rightarrow \neg\theta_j(x)$,

which contradicts consistency of the above since $\theta_j(x)$ can again be replaced with $\theta'_j(x)$ by formula (3.18). By Item (ii) of Claim 1, r can be extended to be a run through $Q \cup \{T_0\}$ by setting $r(T_0) = \{t_0 \in T_0 \mid t_0 \text{ agrees with } t \text{ on the weight formulas}\}$.

Repeating the above steps for each $T \in Q$, $t \in T$ leaves us with a run $r_{t,T}$ through $Q \cup \{T_0\}$ for each such pair t, T . Define a set of runs by taking $R = \{r_{t,T} \mid t \in T, T \in Q\}$. Observe that (T_0, Q, R) is a quasi-model candidate: for each $T \in Q, t \in T$, we have $t \in r_{t,T}(T)$; for each $t \in T_0$, Item (iii) of Claim 1 implies that there is a type $t' \in T'$ for some $T' \in Q$ that agrees on the weight formulas with t , thus $t \in r_{t',T'}(T_0)$. Moreover, we obtain a consistent formula of the form

$$\begin{aligned} & \bigwedge_{T \in Q} w(\chi(T)) > 0 \wedge \sum_{T \in Q} w(\chi(T)) = 1 \wedge \\ & \bigwedge_{r \in R} \left(\bigwedge_{\theta \in r} \theta'(x_r) \wedge \bigwedge_{S \in Q, s \in r(S)} w(\chi(S) \wedge s(x_r)) > 0 \wedge \right. \\ & \quad \left. \bigwedge_{S \in Q} w(\chi(S)) = \sum_{s \in r(S)} w(\chi(S) \wedge s(x_r)) \right) \end{aligned} \quad (3.21)$$

where $\chi(T_0)$ and existential quantification of the variables $x_r, r \in R$ is omitted (possible as we study consistency). Let ϑ' be obtained from (3.21) by replacing each $w(\chi(T))$ with x_T and each $w(\chi(T) \wedge t(x_r))$ in some conjunct for $r \in R$ with $x_{r,t,T}$. Note that ϑ' is a formula in the language of ordered fields which is, in fact, equivalent to $\mathcal{E}(Q, R)$ plus positivity of the solution. Consistency of (3.21) implies that ϑ' is satisfiable in the theory of ordered fields. Therefore, ϑ' is satisfiable in some ordered field F . By the Artin-Schreier theorem [7] in the real-closure of F , thus also in the real numbers. This implies that $\mathcal{E}(Q, R)$ has a positive solution over the reals, that is, (T_0, Q, R) is a quasi-model for φ_0 .

3.5 Decidability and Complexity

Theorem 3.11 reduces satisfiability in monodic ProbFO to satisfiability in FO (in the disguise of checking realizability) and solving systems of polynomial inequalities over the reals. In the following, we use this observation to establish decidability results for fragments of monodic ProbFO that are obtained by restricting its FO part to a decidable FO fragment such as the guarded or the two-variable fragment. We also derive complexity results, which in some cases are tight. Given that we aim at *maximal* decidable fragments of monodic ProbFO, complexities tend to be high. For a fragment \mathcal{L} of FO, let *monodic Prob* \mathcal{L} be the fragment of monodic ProbFO that consists of all formulas φ such that, for all $\psi \in \text{sub}(\varphi)$, the FO formula $\bar{\psi}$ belongs to \mathcal{L} . To warm up, we consider the finite model property (FMP). Recall that, by Theorem 3.11, even full monodic ProbFO has

the FMP regarding the number of worlds. Here, we thus mean the number of domain elements.

Theorem 3.18. *For all FO fragments \mathcal{L} , monodic Prob \mathcal{L} has FMP iff \mathcal{L} has FMP.*

Theorem 3.18 is a direct consequence of the proof of Theorem 3.11. In the “if”-direction of that proof, we combine FO structures that witness realizability of world types. If \mathcal{L} has the finite model property, we can choose these structures to be finite. Then, the resulting probabilistic structure is also finite.

Based on quasi-models, transfer of decidability is also easy to establish. We say that *realizability is decidable in \mathcal{L}* if it is decidable whether a given world type T formulated in monodic Prob \mathcal{L} is realizable.

Theorem 3.19. *If realizability is decidable in the FO fragment \mathcal{L} , then so is satisfiability in monodic Prob \mathcal{L} .*

Theorem 3.19 is established by the following algorithm which decides satisfiability of a given Prob \mathcal{L} sentence φ_0 :

1. guess a quasi-model candidate (T_0, Q, R) for φ_0 ;
2. verify that the system $\mathcal{E}(Q, R)$ has a positive solution in \mathbb{R} ;
3. verify that each world type $T \in Q \cup \{T_0\}$ is realizable.

Step 1 is effective since the size of quasi-model candidates is bounded by a computable function in the size of φ_0 , which is analyzed in more detail below. Step 2 is effective because satisfiability of the system of polynomial inequalities $\mathcal{E}(Q, R)$ over the reals is decidable and realizability is decidable by assumption.

Recall that realizability of some world type T formulated in monodic Prob \mathcal{L} is equivalent to satisfiability of $\bar{\chi}(T)$. Moreover, it turns out that for many FO fragments \mathcal{L} , $\bar{\chi}(T)$ is an \mathcal{L} -formula, that is, realizability in \mathcal{L} can be reduced to satisfiability in \mathcal{L} . In particular, it is easy to verify that this is case for the monadic fragment of FO (MonaFO), the guarded fragment (GF) [2], the guarded negation fragment (GNFO) [15], and the two-variable fragment FO₂ [62]. Thus, Theorem 3.19 applies to all these logics.

Corollary 3.20. *Let \mathcal{L} be one of MonaFO, GF, GNFO, FO₂. Then satisfiability in monodic Prob \mathcal{L} is decidable.*

To analyze the complexity of the algorithm from the proof of Theorem 3.19, first note that it suffices to guess a quasi-model candidate (T_0, Q, R) of size at most double exponential in the size of φ_0 . In fact, Q contains at most double exponentially many world types T , and each T contains at most exponentially many types. While R can in principle be larger than double exponential, it suffices to include one run r for each $T \in Q$ and $t \in T$, such

that $t \in r(T)$. Considering for example GF in which satisfiability is 2EXPTIME-complete, we thus obtain a $2\text{NEXPTIME}^{\exists\mathbb{R}, 2\text{EXPTIME}}$ upper bound for satisfiability in monodic ProbGF where the superscripts indicate access to two oracles: one for solving systems of polynomial inequalities over the reals and one for realizability in GF. Note that $\exists\mathbb{R}$ denotes the class of all problems that are reducible in polynomial time to solving the mentioned systems [121], and that it is known that $\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}$.

3.5.1 Improvements

For many FO fragments \mathcal{L} , though, we can improve on the upper bounds obtained in this direct way. First, it is helpful to not consider satisfiability of the exponential size realizability formula $\bar{\chi}(T)$ as a black box. In particular, the regular structure of $\bar{\chi}(T)$ implies that its satisfiability can be decided in time double exponential *in the size of* φ_0 for GF and in space exponential in the size of φ_0 for both MonaFO and FO₂ [77]. This yields a $2\text{NEXPTIME}^{\exists\mathbb{R}}$ upper bound for monodic ProbGF, monodic ProbMonaFO, and monodic ProbFO₂. Second, we identify a general criterion on \mathcal{L} that potentially leads to an exponential improvement of the basic algorithm. We then study the introduced decidable fragments of ProbFO and investigate the applicability of the criterion. We start with the definition of the criterion.

Definition 3.21 (Closure under union of types). *An FO fragment \mathcal{L} is closed under union of types if for each \mathcal{L} -sentence ψ and any two structures \mathfrak{A}_1 and \mathfrak{A}_2 that satisfy the same sentences from $\text{sub}_x(\psi)$, there is a structure \mathfrak{B} such that $\text{tp}(\mathfrak{B}) = \text{tp}(\mathfrak{A}_1) \cup \text{tp}(\mathfrak{A}_2)$.*

Theorem 3.22. *If \mathcal{L} is closed under union of types, then for every satisfiable monodic Prob \mathcal{L} sentence φ_0 , there is a quasi-model (T_0, Q, R) for φ_0 such that any two distinct world types do not agree on sentences.*

Proof. Let φ_0 be a satisfiable monodic Prob \mathcal{L} sentence. By Theorem 3.11, there is a quasi-model (T_0, Q, R) satisfying φ_0 . Let us write $T \equiv_0 T'$ if two sets of types T, T' agree on sentences, that is, for any $t \in T$, $t' \in T'$, we have that $t \equiv_0 t'$. Define a new quasi-model (T_0, Q', R') as follows:

- $Q' = \{\bigcup_{T' \in Q, T \equiv_0 T'} T' \mid T \in Q\}$;
- for each $r \in R$ define $r' \in R'$ by taking $r'(T_0) = r(T_0)$ and for each $T' \in Q'$:

$$r'(T') = \bigcup_{T \in Q, T \equiv_0 T'} r(T).$$

It should be clear that R' is a set of runs through $Q' \cup \{T_0\}$ and that (T_0, Q', R') is a quasi-model candidate.

Observe first that each $T \in Q'$ is realizable as it is a finite union of realizable $T' \in Q$ and, by assumption, \mathcal{L} is closed under union of types. Next, we show that $\mathcal{E}(Q', R')$ has

a positive solution. By assumption, $\mathcal{E}(Q, R)$ has a positive solution x_T^* for every $T \in Q$ and $x_{r,t,T}^*$ for every $r \in R$, $T \in Q$, and $t \in r(T)$. For each $r' \in R'$, fix some an arbitrary run $\bar{r} \in R$ such that $\bar{r}' = r'$ (there might be more than one). Now, define values $y_{T'}^*$ and $y_{r',t,T'}^*$:

$$\begin{aligned} y_{T'}^* &:= \sum_{T \in Q, T \equiv_0 T'} x_T^*; \\ y_{r',t,T'}^* &:= \sum_{T \in Q, T \equiv_0 T'} \sum_{t \in \bar{r}(T)} x_{\bar{r},t,T}^*. \end{aligned}$$

Clearly, all these values are positive. We show that they are also a solution of $\mathcal{E}(Q', R')$.

- The equations under Item 1 of Definition 3.10 are satisfied as \equiv_0 partitions Q and it was satisfied in $\mathcal{E}(Q, R)$.
- For the equations under Item 2 we have:

$$\begin{aligned} \sum_{t \in r'(T')} y_{r',t,T'}^* &= \sum_{t \in r'(T')} \sum_{T \in Q, T \equiv_0 T'} \sum_{t \in \bar{r}(T)} x_{\bar{r},t,T}^* \\ &= \sum_{T \in Q, T \equiv_0 T'} x_T^* = y_{T'}^*. \end{aligned}$$

- For seeing that the equations under Item 3 are satisfied it suffices to note that:

$$\begin{aligned} [\mathbf{w}(\psi(x))]_{r'} &= \sum_{T' \in Q'} \sum_{t \in r'(T'), \psi(x) \in t} y_{r',t,T'}^* \\ &= \sum_{T' \in Q'} \sum_{t \in r'(T'), \psi(x) \in t} \sum_{T \in Q, T \equiv_0 T'} \sum_{t \in \bar{r}(T)} x_{\bar{r},t,T}^* \\ &= \sum_{T \in Q} \sum_{t \in \bar{r}(T), \psi(x) \in t} x_{\bar{r},t,T}^* \\ &= [\mathbf{w}(\psi(x))]_{\bar{r}} \end{aligned}$$

The third equality is the most subtle: observe that the sums in the second line range over all $T \in Q$ and do not count any $T \in Q$ twice. Moreover, only types t with $\psi(x) \in t$ are considered.

This finishes the proof of Theorem 3.22. ■

In order to show the benefits of the proposed improvements, let us define *ProbGF knowledge bases* as ProbGF-formulas $\varphi_1 \wedge \varphi_2$ where φ_1 is variable free and φ_2 is constant free. Observe that formulas of this form appear regularly in the field of knowledge

representation, for example, in description logics, where assertional and terminological knowledge are separated. The proof of the following result exploits that GF knowledge bases are closed under union of types.

Theorem 3.23. *Deciding satisfiability of monodic ProbGF knowledge bases is 2EXPTIME-complete.*

Proof. The lower bound is inherited from satisfiability in GF. For the upper bound, it is not hard to show that GF restricted to formulas of this form is closed under union of types: given two models $\mathfrak{A}_1, \mathfrak{A}_2$ of $\varphi_0 = \varphi_1 \wedge \varphi_2$ that satisfy the same sentences of $\text{sub}_x(\varphi)$, construct a new structure \mathfrak{B} as the disjoint union of \mathfrak{A}_1 and \mathfrak{A}'_2 , where \mathfrak{A}'_2 is obtained from \mathfrak{A}_2 by dropping the interpretation of all constant symbols. Clearly, \mathfrak{B} is a model for φ and realizes precisely the types realized either in \mathfrak{A}_1 or \mathfrak{A}_2 . By Theorem 3.22, it suffices to guess a quasi-model candidate of exponential size. The associated system $\mathcal{E}(Q, R)$ is then also of exponential size and thus the existence of a solution can be checked in space exponential in the size of the input formula φ_0 since $\exists\text{IR} \subseteq \text{PSPACE}$. It remains to verify that every world type is realizable, in time double exponential in the size of φ_0 . ■

Unfortunately, it turns out that none of the other logics mentioned in Corollary 3.20 is closed under union of types.

Proposition 3.24. *FO_2 , MonaFO , GF , and GNFO are not closed under union of types.*

Proof. We start with FO_2 and MonaFO . Consider the sentence

$$\psi = \forall x (\forall y (A(x) \wedge B(y)) \vee \forall y (\neg A(y) \wedge \neg B(x)))$$

which is equivalent to $\forall x (A(x) \wedge B(x)) \vee \forall x (\neg A(x) \wedge \neg B(x))$, that is, ψ states that either all domain elements satisfy A and B or none. It does this in a slightly unorthodox way to ensure that no sentence from $\text{sub}_x(\psi)$ can distinguish the two cases. More precisely, let $\mathfrak{A}_1, \mathfrak{A}_2$ be models of ψ such that A, B are full in \mathfrak{A}_1 and empty in \mathfrak{A}_2 . Then, all the types in \mathfrak{A}_1 contain the formula $\forall y (A(x) \wedge B(y))$, and all types in \mathfrak{A}_2 contain the formula $\forall y (\neg A(y) \wedge \neg B(x))$. Clearly, the formula $\exists x \forall y (A(x) \wedge B(y)) \wedge \exists x \forall y (\neg A(y) \wedge \neg B(x))$ is unsatisfiable; thus, there cannot be a model realizing all types from \mathfrak{A}_1 and \mathfrak{A}_2 .

For GF and GNFO consider the formula

$$\psi' = \exists x S(x, c, d) \wedge \forall xyz S(x, y, z) \Rightarrow (P(x) \Leftrightarrow R(y, z)).$$

Then, define two models $\mathfrak{A}_1 = (A_1, \pi_1)$ and $\mathfrak{A}_2 = (A_2, \pi_2)$ of ψ' :

$$\begin{array}{ll} A_1 = \{a_1, c, d\} & A_2 = \{a_2, c, d\} \\ \pi_1(P) = \{a_1\} & \pi_2(P) = \emptyset \\ \pi_1(R) = \{(c, d)\} & \pi_2(R) = \emptyset \\ \pi_1(S) = \{(a_1, c, d)\} & \pi_2(S) = \{(a_2, c, d)\} \\ \pi_1(c) = c & \pi_2(c) = c \\ \pi_1(d) = d & \pi_2(d) = d \end{array}$$

Observe that the type $t_1 = \text{tp}(\mathfrak{A}_1, a_1)$ contains the formulas $P(x)$ and $S(x, c, d)$ and the type $t_2 = \text{tp}(\mathfrak{A}_2, a_2)$ contains the formulas $\neg P(x)$ and $S(x, c, d)$. It is not hard to verify that $\exists x (P(x) \wedge S(x, c, d)) \wedge \exists x (\neg P(x) \wedge S(x, c, d))$ is not satisfiable. Thus, there cannot be a model realizing all types from \mathfrak{A}_1 and \mathfrak{A}_2 . ■

Note that closure under union of types is closely related to closure under disjoint union – the property that whenever $\mathfrak{A}_1, \mathfrak{A}_2$ are models of a sentence φ , then so is their disjoint union $\mathfrak{A}_1 \uplus \mathfrak{A}_2$. In fact, closure under disjoint union is one way to prove closure under union of types. Thus, the result is not surprising for MonFO and FO₂ as these logics are not closed under disjoint union.

For GF, the problem is the interaction with the constants, and the restriction to ProbGF knowledge bases provided us with a possibility to restrict this interaction. We next suggest an orthogonal approach that is based on strengthening the definition of types by including *all* ground atoms in the set $\text{sub}_x(\varphi_0)$ of subformulas. For this purpose, we introduce the notion of *extended types*. The set $\text{sub}_x(\varphi_0)$ is now defined as:

$$\begin{aligned} \text{sub}_x(\varphi_0) = \text{sub}_0(\varphi_0) \cup \{ \psi\{x/y\} \mid \psi(y) \in \text{sub}_1(\varphi_0) \} \cup \\ \{ \psi(\vec{c}) \mid \psi(\vec{y}) \in \text{sub}(\varphi_0), \vec{c} \subseteq \text{con}(\varphi_0) \}. \end{aligned}$$

An *extended type* is a maximal satisfiable subset of $\text{sub}_x(\varphi_0)$. It is not hard to verify that the quasi-model machinery developed in this chapter, including Theorem 3.22, works when types are replaced with extended types. However, we have

Proposition 3.25. *GF is closed under union of extended types.*

Proof. Take structures $\mathfrak{A}_1 = (A_1, \pi_1), \mathfrak{A}_2 = (A_2, \pi_2)$ that satisfy the same subsentences of φ_0 . We can assume w.l.o.g. that constants are interpreted as themselves, that they are distinct (no equality in the language), and that the domains A_1 and A_2 are disjoint except for the constants. Define $\mathfrak{B} = (B, \pi)$ as the union of \mathfrak{A}_1 and \mathfrak{A}_2 , that is:

- $B = A_1 \cup A_2$, and
- $\pi(R) = \pi_1(R) \cup \pi_2(R)$ for all predicate symbols R ;

- $\pi(c) = c$ for all constant symbols.

We show the following by induction on the structure of φ :

Claim 1. For all subformulas $\varphi(\vec{x})$ of φ_0 and for all tuples $\vec{a} \subseteq A_i, i \in \{1, 2\}$ we have:

$$\mathfrak{B} \models \varphi[\vec{a}] \iff \mathfrak{A}_i \models \varphi[\vec{a}].$$

Proof of Claim 1. We assume that formulas of GF are built from negation, conjunction, and guarded existential quantification $\exists \vec{y} \alpha(\vec{x}, \vec{y}) \wedge \psi(\vec{x}, \vec{y})$. The guard atom α is allowed to be an equality atom.

For the induction base, assume $\varphi = R(\vec{t})$, where \vec{t} involves free variables \vec{x} that are instantiated by the tuple \vec{a} . The “ \Leftarrow ”-direction is clear by construction. For the “ \Rightarrow ”-direction, we have: $\mathfrak{B} \models R(\vec{t})[\vec{a}]$ implies $\mathfrak{B} \models R(\vec{b})$ where \vec{b} is obtained from \vec{t} by replacing the free variables with the corresponding values from \vec{a} . Thus, $\vec{b} \in \pi(R)$ and $\vec{b} \in \pi_i(R)$ for some i , hence $\mathfrak{A}_i \models R(\vec{b})$. If \vec{b} involves non-constants, all these have to come from A_i , so we are done. If not, that is $\vec{b} \subseteq \text{con}(\varphi_0)$, then $R(\vec{b})$ is a sentence in $\text{sub}_x(\varphi_0)$. As \mathfrak{A}_1 and \mathfrak{A}_2 satisfy the same sentences, we also have $\mathfrak{A}_{3-i} \models R(\vec{b})$, which finishes the proof of the induction base.

The cases for negation and conjunction are immediate consequences of the hypothesis. For existential quantification let $\varphi(\vec{x}) = \exists \vec{y} \alpha(\vec{x}, \vec{y}) \wedge \psi(\vec{x}, \vec{y})$. Consider first the “ \Leftarrow ”-direction. Clearly, \vec{a} can be extended to a tuple \vec{b} in \mathcal{A}_i such that $\mathfrak{A}_i \models \alpha(\vec{b})$ and $\mathfrak{A}_i \models \psi(\vec{b})$. By induction hypothesis, $\mathfrak{B} \models \alpha(\vec{b}) \wedge \psi(\vec{b})$. Thus, we have $\mathfrak{B} \models \varphi[\vec{a}]$. For the “ \Rightarrow ”-direction, we distinguish two cases:

- $\alpha(\vec{x}, \vec{y})$ is an equality $y = y$, i.e., $\varphi = \exists y y = y \wedge \psi(y)$ is a sentence. Thus, there is some a such that $\mathfrak{B} \models \psi[a]$. Assume without loss of generality that a is in the domain of \mathfrak{A}_1 . By induction hypothesis, we have $\mathfrak{A}_1 \models \psi[a]$ and thus $\mathfrak{A}_1 \models \varphi$. As φ is a sentence, we also have $\mathfrak{A}_2 \models \varphi$.
- $\alpha(\vec{x}, \vec{y})$ is an atom $R(\vec{x}, \vec{y})$. By the guard atom, \vec{a} can be extended to a tuple \vec{b} such that $\mathfrak{B} \models R(\vec{b}) \wedge \psi(\vec{b})$. Since $\mathfrak{B} \models R(\vec{b})$, we have $\vec{b} \subseteq A_i$ for at least one i . For all i with $\vec{b} \subseteq A_i$, Induction hypothesis yields $\mathfrak{A}_i \models R(\vec{b})$ and $\mathfrak{A}_i \models \psi(\vec{b})$. Hence, we obtain $\mathfrak{A}_i \models \varphi[\vec{a}]$.

This finishes the proof of Claim 1.

Claim 2. $\text{tp}(\mathfrak{B}) = \text{tp}(\mathfrak{A}_1) \cup \text{tp}(\mathfrak{A}_2)$.

“ \subseteq ”: Let $t \in \text{tp}(\mathfrak{B})$, that is, for some $a \in B$ we have for all $\psi(x) \in t$ that $\mathfrak{B} \models \psi[a]$. If $a \in A_i$, then Claim 1 implies $\mathfrak{A}_i \models \psi[a]$ for all $\psi(x) \in t$ and thus $t = \text{tp}(\mathfrak{A}_i, a)$.

“ \supseteq ”: Assume $t \in \text{tp}(\mathfrak{A}_i)$, that is, for some $a \in A_i$ we have for all $\psi(x) \in t$ that $\mathfrak{A}_i \models \psi[a]$. By Claim 1, also $\mathfrak{B} \models \psi[a]$ for all $\psi(x) \in t$, and thus $t = \text{tp}(\mathfrak{B}, a)$.

This finishes the proof of Claim 2 and thus the proof of the Proposition. ■

Notice that the size of $\text{sub}_x(\varphi_0)$ is not polynomial in $|\varphi_0|$ anymore, but exponential in the maximal arity r of predicate symbols occurring in φ : subformulas of φ_0 have at most r free variables as all free variables need to be guarded by some *atom*. Thus, we cannot improve on the basic algorithm for full GF as the number of types is double exponential now. However, if we restrict our attention to predicates with *bounded arities*, we get as a consequence of Theorem 3.22 (as mentioned: lifted to extended types) the following improved complexity bounds.

Corollary 3.26. *Satisfiability in monodic ProbGF is in $\text{NEXPTIME}^{\exists\mathbb{R}}$ when the arity of predicates is bounded. It is NEXPTIME -complete when only linear weight formulas are allowed.*

Proof. The $\text{NEXPTIME}^{\exists\mathbb{R}}$ upper bound follows from the fact that it suffices to guess an exponentially sized quasi-model since realizability can be checked in exponential time. Observe that for restriction to linear weight formulas, $\mathcal{E}(Q, R)$ is in fact an exponentially sized system of *linear* inequalities and can thus be solved in exponential time. The NEXPTIME lower bound follows from the fact that the NEXPTIME -hard modal description logic $\text{S5}_{\mathcal{ALC}}$ [52] is contained in this fragment. ■

We finish this section by noting that Theorem 3.22 does not imply any lower bounds, that is, it does not rule out that, e.g., satisfiability for full ProbGF is possible in 2EXPTIME . In particular, we were aiming at a *general* criterion for improving the complexity of the basic algorithm. Surprisingly, this general criterion yields tight upper bounds in some relevant cases. In this chapter, we leave the precise complexities of satisfiability in ProbGF, ProbGNFO, ProbMonaFO, and ProbFO₂ for future work, and instead focus on the implications of the above results for probabilistic description logics as recently introduced in [101].

3.6 Connection to Probabilistic Description Logics

Let us revisit probabilistic description logics (ProbDLs) and the results obtained for them [101] in the light of the developments of this chapter. We consider probabilistic structures with *rigid constants* in this section, since the semantics for ProbDLs features rigid interpretation of the individual names. We have argued after (the proof of) Theorem 3.11 that the quasi-model machinery can be adapted to this setting. Also, it is not hard to verify that the remaining results can be lifted as well.

We study the relationship of ProbDLs to well-behaved fragments of ProbFO by extending the well-known translation from \mathcal{ALC} into first-order logic, see for example [12], to a translation of Prob \mathcal{ALC} to ProbFO. More specifically, we provide translation functions tr and tr_z , $z \in \{x, y\}$ from Prob \mathcal{ALC} -concepts, TBoxes, and ABoxes into

ProbFO-formulas by taking:

$$\begin{aligned}
 \text{tr}_z(A) &= A(z) & \text{tr}_z(\neg C) &= \neg \text{tr}_z(C) \\
 \text{tr}_z(C \sqcap D) &= \text{tr}_z(C) \wedge \text{tr}_z(D) & \text{tr}_z(\exists r.C) &= \exists \bar{z} (R(z, \bar{z}) \wedge \text{tr}_{\bar{z}}(C)) \\
 \text{tr}_z(P_{\sim p}C) &= \text{w}(\text{tr}_z(C)) \sim p & \text{tr}_z(\exists P_{\sim p}r.C) &= \exists \bar{z} (\text{w}(R(z, \bar{z})) \sim p \wedge \text{tr}_{\bar{z}}(C)) \\
 \text{tr}(\mathcal{T}) &= \bigwedge_{C \sqsubseteq D \in \mathcal{T}} \text{tr}(C \sqsubseteq D) & \text{tr}(C \sqsubseteq D) &= \forall x (\text{tr}_x(C) \rightarrow \text{tr}_x(D)) \\
 \text{tr}(C(a)) &= \text{tr}_x(C)[x/a] & \text{tr}(r(a, b)) &= R(a, b) \\
 \text{tr}(\neg \mathcal{A}) &= \neg \text{tr}(\mathcal{A}) & \text{tr}(\mathcal{A} \wedge \mathcal{A}') &= \text{tr}(\mathcal{A}) \wedge \text{tr}(\mathcal{A}') \\
 \text{tr}(P_{\sim p}\mathcal{A}) &= \text{w}(\text{tr}(\mathcal{A})) \sim p
 \end{aligned}$$

where $\bar{x} = y$ and $\bar{y} = x$. The translation function tr can be used to reduce consistency checking Prob \mathcal{ALC} to satisfiability in ProbFO as follows.

Proposition 3.27. *A knowledge base $(\mathcal{T}, \mathcal{A})$ is consistent iff the ProbFO formula*

$$\varphi_{\mathcal{T}, \mathcal{A}} = \text{tr}(\mathcal{T}) \wedge \text{tr}(\mathcal{A}) \wedge \text{w}(\text{tr}(\mathcal{T})) = 1$$

is satisfiable.

The proof is straightforward and relies on the close correspondence of probabilistic interpretations (for Prob \mathcal{ALC}) and probabilistic structures (for ProbFO). The crucial observation for both directions is that we can restrict attention to satisfiability in a world of probability 0. The additional $\text{tr}(\mathcal{T})$ is necessary since $\text{w}(\text{tr}(\mathcal{T})) = 1$ does not cover worlds with probability 0.

Let us take a closer look at the formula $\varphi_{\mathcal{T}, \mathcal{A}}$. By definition of the translation functions, $\varphi_{\mathcal{T}, \mathcal{A}}$ is a formula in ProbGF, but not in *monodic* ProbGF due to the translation of probabilistic roles. Hence, in general, the translation results in a formula outside the fragments that we identified as “well-behaved”. This can be taken as justification of why the decidability status of the problem is hard to settle. Conversely, we can explain the decent complexity results mentioned in the preliminaries by the following:

- For Prob \mathcal{ALC}_c knowledge bases $(\mathcal{T}, \mathcal{A})$, $\varphi_{\mathcal{T}, \mathcal{A}}$ is a sentence in monodic ProbGF. As the predicate arity is bounded by 2 and $\varphi_{\mathcal{T}, \mathcal{A}}$ involves only *linear* weight formulas, Corollary 3.26 yields a NEXPTIME-upper bound. To give an intuitive explanation for the EXPTIME upper bound from [101], we can argue as follows. Note that, except from possible subformulas of $\text{tr}(\mathcal{A})$, there is only one occurrence of a (top-level) weight formula $\text{w}(\psi) \sim p$ with ψ a sentence, namely $\text{w}(\text{tr}(\mathcal{T})) = 1$. In a sense, this formula gives rise to *precisely one* world type. Thus, the guessing step in the general algorithm can be avoided.

- For $\text{Prob}\mathcal{ALC}_{01}$ knowledge base $(\mathcal{T}, \mathcal{A})$, $\varphi_{\mathcal{T}, \mathcal{A}}$ is a sentence in ProbFO_{01} which we found to be a well-behaved fragment of ProbFO (also for rigid constants). Again, this observation can be viewed as an explanation of the (relatively low) complexity of concept satisfiability in $\text{Prob}\mathcal{ALC}_{01}$.

Thus, we have shown that the computational behavior of ProbDLs can be explained by embedding them into ProbFO. Most notably, monodicity explains the low computational complexity of $\text{Prob}\mathcal{ALC}_c$, the fragment with only probabilistic concepts.

3.7 Conclusion and Outlook

In this first chapter, we have analyzed the reasons for the disastrous computational behaviour of ProbFO and we have shown that, unlike other natural restrictions that fail to establish recursive enumerability and decidability, *monodicity* is able to tame ProbFO computationally. We introduced the technical framework of *quasi-models* and proved its usefulness by showing axiomatizability and decidability transfer theorems. We were even able to identify fragments where our general decision procedure behaves optimal, that is, it provides a tight upper bound. We thus believe that monodic ProbFO lays a promising foundation for identifying decidable and relevant probabilistic logics for computer science. As detailed in this chapter, monodic ProbFO in the presented form turned out to be a significant generalization of some probabilistic description logics recently introduced [101]. In particular, we can explain the decent computational behavior of some ProbDLs by monodicity. There are several interesting options for future research.

Open technical problems

Let us start with the technical problems that were left open in this chapter. It remains to determine the precise complexity for several logics such as full monodic ProbGF (no restrictions on constants or arities) or even monodic ProbGNFO. We believe this requires specialized techniques for every single logic, in contrast to the general approach from Section 3.5.1. Second, we argued that *equality* imposes severe technical problems on lifting our approach to $\text{ProbFO}^=$. Interestingly, in the case of temporal FO, the monodic fragment with equality is not recursively enumerable even when constant symbols are disallowed [131]. While also monodic $\text{ProbFO}^=$ is not recursively enumerable by Theorem 3.8, the proof of this result crucially relies on constant symbols and it remains open whether validity in $\text{ProbFO}^=$ without constant symbols is recursively enumerable.

Independences

An interesting direction for further research is to enrich monodic ProbFO with additional expressive power that enables complex and succinct statements about independence. This

would in particular be useful to enable various applications in AI, where independence and succinct representations thereof are in the focus of attention.

Combination with Statistical knowledge

Another important extension to be investigated is the combination of statistical and subjective probabilities in a probabilistic FO logic. A basic version of ProbFO that combines both kinds of probability was considered by Halpern under the name *type-3 ProbFO* [67]. The statistical component is realized in the semantics by adding a *discrete distribution* η over the domain, that is, type-3 probabilistic structures are tuples $\mathfrak{M} = (D, W, \pi, \mu, \eta)$. In this logic, we can write statistical formulas like $w_x(\text{Flies}(x) \mid \text{Bird}(x)) \geq 0.9$, expressing that 90% of all birds fly, or mix statistical and subjective formulas:

$$w_x(\text{Wealthy}(x) \mid \text{Swiss}(x)) \geq 0.6 \wedge \text{Swiss}(\text{albert}) \wedge w(\text{Wealthy}(\text{albert})) = .01. \quad (3.22)$$

Let us remark that in the advocated combination, the two forms of uncertainties are largely ‘independent’. In particular, the above sentence (3.22) is satisfiable. Intuitively, although it is known that at least 60 % of the Swiss population is wealthy and **albert** is Swiss, it is consistent to believe that **albert** is probably not wealthy. Thus, the statistics do not have immediate influence on the subjective probabilities.

From a practical perspective, this independence is often unsatisfactory; in contrast, the *transfer of statistical knowledge to degrees of belief* is a prominent application in artificial intelligence [95, 14, 92]. Let us illustrate this by picking up the Swiss/wealth example from above. In particular, from knowing that 60 % of the population is wealthy and **albert** being a Swiss, we would like to *conclude* a degree of belief of 0.6 that he is rich. Thus, the sentence in (3.22) should be unsatisfiable. This is commonly known as *direct inference* [117] which can be justified by Laplace’s *principle of indifference* [98]: if we look at all Swiss people without further distinction, we should not have different beliefs about their monetary status. This is only possible if we believe of everybody that she is wealthy with probability 60 %.

Other related reasoning patterns in this context are *irrelevance reasoning* and *reference class reasoning*. The former refers to the fact that adding more (unrelated) knowledge such as **Tall(albert)** – the fact that **albert** is tall – should not destroy our belief of 0.6 that **albert** is rich. In contrast, for the latter assume that the knowledge base additionally contains a condition for more wealth such as working for the bank UBS. We would then like to have that

$$\begin{aligned} w_x(\text{Wealthy}(x) \mid \text{Swiss}(x)) \geq 0.6 \wedge \text{Swiss}(\text{albert}) \wedge \text{worksAt}(\text{albert}, \text{UBS}) \wedge \\ w_x(\text{Wealthy}(x) \mid \text{Swiss}(x) \wedge \text{worksAt}(x, \text{UBS})) \geq 0.9 \end{aligned}$$

implies that the subjective belief that **albert** is rich is 90 %. Thus, the most specific reference class for an individual is preferred, which in this case is being Swiss and working

UBS. There has quite some work aiming to devise a general scheme capturing some or all of the mentioned patterns, also in the context of first-order probabilistic logics, please consult [14, 92] and the references therein, but so far there is no complete solution. Note that incorporating *any* of the aforementioned reasoning patterns renders the logic non-monotonic, in contrast to the logics considered in this chapter and also plain type-3 ProbFO.

It would be interesting to see whether our framework of monodic ProbFO can be extended to capture plain type-3 ProbFO, possibly by identifying a suitable condition on the statistical weight formulas, similar to monodicity. Having established this, it would be a fundamental contribution to incorporate reasoning mechanisms like direct inference, irrelevance, or reference class reasoning.

Notably, despite the seeming independence of the dimensions, decidability of monodic versions of (plain!) type-3 ProbFO turns out to be non-trivial to establish, and it is currently not clear how to extend the quasi-model machinery to this logic. We illustrate the difficulties with a somewhat unexpected effect. Recall that a satisfiable monodic ProbFO formula can be satisfied in a model with finitely many worlds and that the number of worlds is determined by the solution of the system of inequalities $\mathcal{E}(Q, R)$, which in turn depends on the subjective probabilities given in the input formula, see further the proof of Theorem 3.11. Now consider the formula

$$\forall x \mathbf{w}(P(x)) > 0 \wedge \mathbf{w}(w_x(P(x)) \leq p) = 1,$$

which says that every domain element satisfies P in some world with positive probability, but in every world only a proportion of at most p of the domain elements satisfies P . This enforces the existence of at least $1/p$ worlds. Thus, the number of worlds depends also on the statistical probabilities, which shows some non-trivial technical interplay between statistical and subjective probabilities, although at first sight they seem independent. Thus, we leave an adaptation of the quasi-model machinery to type-3 ProbFO and its extensions as a challenging open research objective.

Combinations with S5

Note that ProbFO₀₁ can be viewed as the product of first-order logic with the modal logic **S5**. We have shown that this combination is well-behaved in the sense that validity is recursively enumerable. There has been some work on combinations of first-order fragments with **S5** showing that often the complexity of satisfiability not much worse than the complexity in the FO fragment [6, 52], and we also contribute a positive result in the next chapter. However, despite a recursive enumerable validity problem in the ‘base’ logic ProbFO₀₁, there are differences to the monodic framework. For example, an analog of Theorem 3.19 cannot be proved since the two-variable fragment of ProbFO₀₁ is undecidable [52, 53]. Thus, it is interesting future work to better understand the complexity of such combinations.

4 Subjective Uncertainty in \mathcal{EL}

In the previous chapter, we have shown that unfortunately, but not surprisingly given their expressiveness, the identified decidable fragments exhibit high computational complexity. Some slightly better behaving exceptions are members of the recently introduced family of *probabilistic description logics* (*ProbDLs*) which can be viewed as fragments of monodic ProbFO. For instance, reasoning in Prob \mathcal{ALC}_c , the restriction of Prob \mathcal{ALC} to probabilistic concepts, c.f. Section 3.6, is EXPTIME-complete and thus not harder than in the base logic \mathcal{ALC} [101]. Motivated by the fact that \mathcal{EL} is a fragment of \mathcal{ALC} offering polynomial time reasoning services, we will focus on the \mathcal{EL} -fragment of Prob \mathcal{ALC} with the hope to find well-behaved probabilistic description logics. Syntactically, Prob \mathcal{EL} is obtained from Prob \mathcal{ALC} by dropping the constructor for negation $\neg C$, and thus also disjunction \sqcup and universal restrictions $\forall r.C$. Let us note that some of the examples that we used so far are actually in Prob \mathcal{EL} . For instance, we can express that a gastric ulcer is an ulcer located at the stomach by the concept definition

$$\text{GastricUlcer} \equiv \text{Ulcer} \sqcap \exists \text{locatedAt.Stomach},$$

or describe patients having a disease that is infectious with probability at least 0.25, by

$$\text{Patient} \sqcap \exists \text{finding.}(\text{Disease} \sqcap P_{>0.25} \text{Infectious}).$$

Recall that in \mathcal{EL} -based languages the studied reasoning problem is typically subsumption, which we will adopt here as well. There are partial results on the complexity of subsumption in Prob \mathcal{EL} [101]:

- EXPTIME-hardness for some fragments of Prob \mathcal{EL} , such as the one allowing for the two probabilistic operators $P_{>0}$ and $P_{>0.4}$ applied to concepts;
- a PTIME algorithm for the fragment allowing for probabilistic operators $P_{>0}$ and $P_{=1}$ applied to concepts;
- PSPACE-hardness for Prob \mathcal{EL}_{01} , the fragment from the previous point with the probabilistic operators also applied to roles. Moreover, a 2EXPTIME-upper bound is inherited from Prob \mathcal{ALC}_{01} , the corresponding \mathcal{ALC} -fragment.

The purpose of this chapter is to provide a more complete picture of subsumption in probabilistic variants of \mathcal{EL} . More specifically, our aim is to find answers to the following questions:

- Are there tractable fragments except for the one in the second item above? Or can we generalize the EXPTIME-hardness from the first item?
- What is the precise complexity of the fragment Prob \mathcal{EL} when also probabilistic roles are allowed? In particular, can we close the complexity gap for Prob \mathcal{EL}_{01} ?

Related Work

In the previous chapter, we have already related the underlying logic ProbFO to other proposals in the literature. Also, in the context of probabilistic description logics, there has been a large number of proposals of different spirits, see for example [93, 101, 100, 82, 31, 99, 109, 42] and the references therein. Here, we will restrict our attention to some proposals of probabilistic logics that are based on tractable logics, and point out rather clearly the commonalities and differences.

Let us start with P-CLASSIC, a proposal by Koller *et al.* from 1997 [93]. It is a probabilistic version of the description logic CLASSIC that additionally uses Bayesian networks to express uncertainty about properties of some individual. For instance, one can specify a distribution over the number of children of a person. It is shown that reasoning in this logic is not more difficult than reasoning in Bayesian networks; in particular, polynomial time when restricted to tractable variants of Bayesian networks. This work differs from Prob \mathcal{EL} in two crucial aspects. First, the semantics is based on a distribution on the domain, that is, probabilities are interpreted *statistically* as opposed to *subjectively* in Prob \mathcal{EL} . Second, in P-CLASSIC as in Bayesian networks, one has to completely specify the distribution and has implicit independence assumptions.

Log-linear description logics [109] are a combination of \mathcal{EL} and the framework of Markov Logic [118]. Hence, it shares many properties with Markov logic. Most importantly, probabilities are interpreted in a subjective way and a knowledge base specifies a single distribution over fixed possible worlds. Weights (the means for expressing uncertainty in Markov logics) are attached to concept inclusions $C \sqsubseteq D$. Thus, a world is characterized by a set of concept inclusions being true. This is largely orthogonal to our semantics as Prob \mathcal{EL} supports probabilistic concepts and roles and no probabilistic concept inclusions. We deliberately dispense with such concept inclusions as we are interested in the uncertainty about specific individuals. Reasoning in this logic is NP-hard.

Another instance of Markov logic is Tractable Markov Logic (TML) by Domingos and Webb from 2012 [42]. The language allows to express statements such as probabilistic inheritance hierarchies. Also being a Markov logic, TML has similar properties as the mentioned log-linear description logics, that is, probabilities are interpreted subjectively and every knowledge base encodes precisely one distribution over the worlds. The difference, however, is that TML allows for polynomial time probabilistic reasoning.

The work perhaps closest to ours is by Finger *et al.* who study satisfiability in \mathcal{EL}

with probabilistic ABoxes [47] and adopt the same semantics based on ProbFO as we do. In contrast to us, they consider standard (non-probabilistic) TBoxes and probabilistic ABoxes, which are conjunctions of statements of the form $P_{\sim p_i}(\mathcal{A}_i)$. They give a non-deterministic polynomial time algorithm for deciding satisfiability of such knowledge bases.

Contribution and Structure of the Chapter

The purpose of this chapter is to establish a more complete picture of subsumption in Prob \mathcal{EL} . In Section 4.1, we start with introducing the syntax and semantics of probabilistic \mathcal{EL} , as well as the relevant reasoning problems of subsumption. Then, the main part of this chapter is divided into two parts. In the first part, Section 4.2, we consider fragments of Prob \mathcal{EL}_c , that is, fragments in which probabilistic operators can only be applied to concepts. In the second part, Section 4.3, we drop this restriction and study full Prob \mathcal{EL} , in which probabilistic operators can be applied to both concepts and roles, but not to concept inclusions. This separation is motivated both by the results from the last section and by the results known from ProbDLs, where it was shown that probabilistic roles tend to increase the complexity [101].

As mentioned above, for Prob \mathcal{EL}_c it was known that some concrete combinations of probability constructors such as $P_{>0}$ and $P_{>0.4}$ lead to intractability (in fact, EXPTIME-completeness) of subsumption while a restriction to the probability values zero and one does not [101]. We prove the much more general result that the extension of \mathcal{EL} with *any* single concept constructor $P_{\sim p}$, where $\sim \in \{<, \leq, =, \geq, >\}$ and $p \in (0, 1)$, results in EXPTIME-hardness of subsumption relative to general TBoxes; containment in EXPTIME is inherited from the EXPTIME-upper bound of the same problem for Prob \mathcal{ALC}_c -TBoxes. The tool for proving these lower bounds is showing *non-convexity* of each logic. Intuitively, a logic is non-convex, if it can express disjunction. Note that none of the logics we consider in this chapter provides disjunction \sqcup as a connective. Hence, in our context non-convexity is witnessed by a TBox \mathcal{T} and concepts C, D_1, \dots, D_k such that $\mathcal{T} \models C \sqsubseteq D_1 \sqcup \dots \sqcup D_k$ but *not* $\mathcal{T} \models C \sqsubseteq D_i$ for any i . Non-convexity can then be used to reduce from satisfiability in \mathcal{ALC} . This is a general observation that was exploited before, see for example [9, 5]. Inspired by the fact that many biomedical ontologies such as SNOMED CT are *classical TBoxes*, that is, sets of concept definitions $A \equiv D$ with A atomic, we then show that probabilities other than zero and one *can* be used without losing tractability in classical TBoxes for the cases $\sim \in \{>, \geq\}$. More precisely, subsumption in Prob \mathcal{EL} is tractable when only the constructors $P_{\sim p}$ and $P_{=1}$ are admitted, for any (single!) choice of $\sim \in \{\geq, >\}$ and $p \in (0, 1)$. Moreover, we show that all the logics obtained in this way actually coincide for all possible choices of p .

For full Prob \mathcal{EL} , where probabilities can be applied to both concepts and roles, it is not hard to see that subsumption relative to general TBoxes exhibits same complexity as concept satisfiability in Prob \mathcal{ALC} . For the fragment Prob \mathcal{EL}_{01} , where only probability

values 0 and 1 are allowed, it was known that subsumption is in 2EXPTIME and PSPACE-hard [101]. It is interesting to note that, so far, any two-dimensional extension of \mathcal{EL} turned out to have the same complexity as the corresponding extension of the expressive DL \mathcal{ALC} , see e.g. [5]. Since subsumption in $\text{Prob}\mathcal{ALC}_{01}$ is 2EXPTIME-complete, it was tempting to conjecture that the same holds for $\text{Prob}\mathcal{EL}_{01}$. We show that this is not the case by establishing a matching PSPACE upper bound for subsumption in $\text{Prob}\mathcal{EL}_{01}$. This also implies PSPACE-completeness for the two-dimensional DL $\mathbf{S5}_{\mathcal{EL}}$, in sharp contrast to the 2EXPTIME-completeness of $\mathbf{S5}_{\mathcal{ALC}}$ [6]. We conclude by showing maximality of the fragment in the following sense: whenever adding another probabilistic operator $P_{\sim p}$ on concepts or roles with $p \in (0, 1)$, reasoning becomes 2EXPTIME-hard.

4.1 Syntax and Semantics of $\text{Prob}\mathcal{EL}$

The logic $\text{Prob}\mathcal{EL}$ is obtained from $\text{Prob}\mathcal{ALC}$ in the same way as \mathcal{EL} is obtained from \mathcal{ALC} , that is, by dropping the constructor \neg for negation and thus also the abbreviations disjunction \sqcup and universal restrictions $\forall r.C$. Thus, the semantics of $\text{Prob}\mathcal{EL}$ is the same as for $\text{Prob}\mathcal{ALC}$. For the sake of completeness, let us state that $\text{Prob}\mathcal{EL}$ -concepts are formed using the syntax rule

$$C, D ::= \top \mid A \mid \exists r.C \mid C \sqcap D \mid P_{\sim p}C \mid \exists P_{\sim p}r.C,$$

where A is a concept name, r is a role name, \sim ranges over $\{<, \leq, =, \geq, >\}$ and $p \in [0, 1]$. We refer with $\text{Prob}\mathcal{EL}_c$ to the fragment of $\text{Prob}\mathcal{EL}$ which allows only for probabilistic concepts. As example, the SNOMED CT concept ‘animal bite by potentially rabid animal’ can be expressed as

$$\text{Bite} \sqcap \exists \text{by}.(\text{Animal} \sqcap P_{>0.5} \exists \text{has}. \text{Rabies}),$$

which is clearly a $\text{Prob}\mathcal{EL}_c$ -concept. An example for a non- $\text{Prob}\mathcal{EL}_c$ -concept is ‘possible viral origin’ which can be expressed using a probabilistic role as $\exists P_{>0} \text{origin}. \text{Viral}$. We will also consider the restriction of $\text{Prob}\mathcal{EL}$ to a subset of probabilistic constructors; for instance, if we allow $\sim p$ to be only <0.3 and ≥ 0.2 we denote this with $\text{Prob}\mathcal{EL}^{<0.3; \geq 0.2}$. The fragment of $\text{Prob}\mathcal{EL}$ allowing for operators $P_{>0}$ and $P_{=1}$ both to concepts and roles is referred to with $\text{Prob}\mathcal{EL}_{01}$.

The central reasoning task for (extensions of) \mathcal{EL} is *subsumption checking*. In the context of ProbDLs there are two natural notions of subsumption: *unrestricted* subsumption and *positive* subsumption.

Definition 4.1 (Unrestricted and Positive Subsumption). *A concept C is unrestrictedly subsumed by D relative to a TBox \mathcal{T} if for all models \mathcal{I} of \mathcal{T} we have $\mathcal{I} \models C \sqsubseteq D$. A concept C is positively subsumed by D relative to a TBox \mathcal{T} if for all probabilistic models $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} and all worlds $w \in W$ with $\mu(w) > 0$*

we have $C^{\mathcal{I},w} \subseteq D^{\mathcal{I},w}$. We denote the former with $\mathcal{T} \models^0 C \sqsubseteq D$ and the latter with $\mathcal{T} \models^+ C \sqsubseteq D$.

Intuitively, unrestricted subsumption captures subsumptions that are *logically implied*, that is, which are valid in all worlds. Thus, it can be seen as the result of transferring the notion of subsumption from standard DLs to probabilistic DLs in a straightforward way. Positive subsumption, in contrast, is taking into account only worlds with positive probability, and is thus about subsumptions that are *certain*. For example, when \mathcal{T}_\emptyset is the empty TBox, then $\mathcal{T}_\emptyset \not\models^0 P_{=1}A \sqsubseteq A$, but we can only have $d \in (P_{=1}A)^{\mathcal{I},v} \setminus A^{\mathcal{I},v}$ when $\mu(v) = 0$, thus non-subsumption is only witnessed by worlds that are certainly not the actual world. Consequently, $\mathcal{T}_\emptyset \models^+ P_{=1}A \sqsubseteq A$. Despite the semantical differences, we argue that we can mutually reduce unrestricted subsumption and positive subsumption.

Lemma 4.2. *Unrestricted and positive subsumption are equivalent under polynomial-time reductions.*

Proof. For reducing positive to unrestricted subsumption, it suffices to observe that

$$\mathcal{T} \models^+ A \sqsubseteq B \quad \text{iff} \quad \mathcal{T} \models^0 P_{>0}(A \sqcap X) \sqsubseteq P_{>0}(B \sqcap X)$$

for a fresh concept name X . The X is needed to ‘mark’ a world witnessing that A is true with positive probability.

For the reduction from unrestricted to positive subsumption, we need some new notation. For a concept C , denote with \hat{C} the concept that is obtained from C by replacing every concept name A or role name r that is not in the scope of some probabilistic operator $P_{\sim p}$ with a fresh name \hat{A} and \hat{r} , respectively. For example, for $C = B \sqcap \exists P_{>0}r.A$ we have $\hat{C} = \hat{B} \sqcap \exists P_{>0}r.\hat{A}$ as the scope of the probabilistic operator $P_{>0}$ is r , not A . Define $\hat{\mathcal{T}}$ as $\{\hat{C} \sqsubseteq \hat{D} \mid C \sqsubseteq D \in \mathcal{T}\}$. It is straightforward to show that

$$\mathcal{T} \models^0 A \sqsubseteq B \quad \text{iff} \quad \mathcal{T} \cup \hat{\mathcal{T}} \models^+ \hat{A} \sqsubseteq \hat{B}.$$

For the “if”-direction, assume $\mathcal{T} \not\models^0 A \sqsubseteq B$ witnessed by a model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} , a world $w \in W$, and an element $d \in \Delta^{\mathcal{I}}$ such that $d \in A^{\mathcal{I},w}$ but $d \notin B^{\mathcal{I},w}$. Define an extension $\mathcal{J} = (\Delta^{\mathcal{I}}, W, (\mathcal{J}_w)_{w \in W}, \mu)$ of \mathcal{I} by choosing a world $\hat{w} \in W$ with $\mu(\hat{w}) > 0$ and taking:

- $\hat{X}^{\mathcal{I},\hat{w}} = X^{\mathcal{I},w}$ for all $X \in \mathbf{N}_C$;
- $\hat{r}^{\mathcal{I},\hat{w}} = r^{\mathcal{I},w}$ for all $r \in \mathbf{N}_R$.

Clearly, we have $d \in \hat{A}^{\mathcal{I},\hat{w}} \setminus \hat{B}^{\mathcal{I},\hat{w}}$. By construction, we have $\mathcal{J} \models \mathcal{T}$; thus it remains to show that $\mathcal{J} \models \hat{\mathcal{T}}$, which is a consequence of the following claim.

Claim 1. For all concepts C , $e \in \Delta^{\mathcal{I}}$: $e \in C^{\mathcal{I},w}$ iff $e \in \hat{C}^{\mathcal{J},\hat{w}}$.

Proof of Claim 1. The proof is by structural induction. For $C = A$ a concept name, it is clear by definition of \mathcal{J} . The case $C = C_1 \sqcap C_2$ is immediate from the induction hypothesis. The case $C = P_{\sim p}D$ follows from the fact that D does not include any concepts of the form \hat{X} . The case $C = \exists r.D$ is as follows:

$$\begin{aligned} e \in (\exists r.D)^{\mathcal{I},w} &\Leftrightarrow \exists e' \in D^{\mathcal{I},w} \wedge (e, e') \in r^{\mathcal{I},w} \\ &\Leftrightarrow \exists e' \in \hat{D}^{\mathcal{J},\hat{w}} \wedge (e, e') \in \hat{r}^{\mathcal{J},\hat{w}} \\ &\Leftrightarrow e \in (\exists \hat{r}.\hat{D})^{\mathcal{J},\hat{w}}. \end{aligned}$$

The case of $C = \exists P_{\sim p}r.D$ is similar. This finishes the proof of the claim.

For the “only if”-direction, assume $\mathcal{T} \cup \hat{\mathcal{T}} \not\models^+ \hat{A} \sqsubseteq \hat{B}$. Thus, there is a model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of $\mathcal{T} \cup \hat{\mathcal{T}}$, a world $w \in W$, and an element $d \in \Delta^{\mathcal{I}}$ such that $d \in \hat{A}^{\mathcal{I},w}$ but $d \notin \hat{B}^{\mathcal{I},w}$. Define an interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \hat{W}, (\mathcal{J}_w)_{w \in \hat{W}}, \hat{\mu})$ by:

- $\hat{W} = W \cup \{\hat{w}\}$ and $\hat{\mu}(v) = \mu(v)$ for all $v \in W$ and $\hat{\mu}(\hat{w}) = 0$;
- $X^{\mathcal{J},v} = X^{\mathcal{I},v}$ and $r^{\mathcal{J},v} = r^{\mathcal{I},v}$ for all $v \in W$;
- $X^{\mathcal{J},\hat{w}} = \hat{X}^{\mathcal{I},w}$ and $r^{\mathcal{J},\hat{w}} = \hat{r}^{\mathcal{I},w}$ for all $X \in \mathbf{N}_C$ and $r \in \mathbf{N}_R$.

Intuitively, \hat{w} is a copy of w where the extension of C is given by $\hat{C}^{\mathcal{I},w}$.

Claim 2. For all concepts C , $e \in \Delta^{\mathcal{I}}$: $e \in \hat{C}^{\mathcal{I},w}$ iff $e \in C^{\mathcal{I},w}$.

Proof of Claim 2. The claim is proved by structural induction. The base case $C = A$ is trivial. The case $C = C_1 \sqcap C_2$ is immediate from the induction hypothesis. The case $C = P_{\sim p}D$ follows from the fact that $\hat{D} = D$ and the added world does not change the probabilities. The case $C = \exists r.D$ is as follows: $e \in (\exists \hat{r}.\hat{D})^{\mathcal{I},w}$ implies that there is some $e' \in \hat{D}^{\mathcal{I},w}$ such that $(e, e') \in \hat{r}^{\mathcal{I},w}$. By construction and induction hypothesis, this is equivalent to $e' \in D^{\mathcal{J},\hat{w}}$ and $(e, e') \in r^{\mathcal{J},\hat{w}}$; this is the case if, and only if $e \in D^{\mathcal{J},\hat{w}}$. The case $C = \exists P_{\sim p}r.D$ is similar. This finishes the proof of the claim.

Clearly, $C^{\mathcal{J},v} \subseteq D^{\mathcal{J},v}$ for all $v \in W$ and $C \sqsubseteq D \in \mathcal{T}$. By the claim, we also have $C^{\mathcal{J},\hat{w}} \subseteq D^{\mathcal{J},\hat{w}}$ for all $C \sqsubseteq D \in \mathcal{T}$. Thus, $\mathcal{T} \models \mathcal{J}$ and it remains to note that $d \in A^{\mathcal{J},\hat{w}} \setminus B^{\mathcal{J},\hat{w}}$, also by the claim. ■

It is not hard to verify that Lemma 4.2 applies to all Prob \mathcal{EL} variants that we consider in this chapter. Thus, it is without loss of generality, to restrict our attention to one of the notions. In the remainder of the chapter, we study positive subsumption, call it only “subsumption” and abbreviate \models^+ with just \models , that is, we write $\mathcal{T} \models A \sqsubseteq B$ instead of $\mathcal{T} \models^+ A \sqsubseteq B$.

4.2 Complexity of Probabilistic Concepts

In this section, we consider fragments of $\text{Prob}\mathcal{EL}_c$, where we drop the constructor $\exists P_{>0}r.C$ and apply probabilistic operators to concepts only. It was shown that subsumption in $\text{Prob}\mathcal{EL}_c^{>0;=1}$ relative to general TBoxes is in PTIME, whereas the same problem is EXPTIME-complete for $\text{Prob}\mathcal{EL}_c^{>0;>0.4}$ [101]. This raises the question which probabilities except 0 and 1 can be admitted in $\text{Prob}\mathcal{EL}_c$ without losing tractability.

4.2.1 Lower bounds

Our first main theorem provides the strong negative result that there is no such value.

Theorem 4.3. *For all $p \in (0, 1)$ and $\sim \in \{\leq, <, =, >, \geq\}$, subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p}$ relative to general TBoxes is EXPTIME-hard.*

Matching upper bounds are an immediate consequence of the fact that each logic $\text{Prob}\mathcal{EL}_c^{\sim p}$ is a fragment of the probabilistic DL $\text{Prob}\mathcal{ALC}_c$ for which subsumption was proved EXPTIME-complete in [101]. Hence, we obtain the following corollary.

Corollary 4.4. *For all $p \in (0, 1)$ and $\sim \in \{\leq, <, =, >, \geq\}$, subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p}$ relative to general TBoxes is EXPTIME-complete.*

In order to prove Theorem 4.3 we require the notion of (non-)convexity.

Definition 4.5 (Convex). *A logic \mathcal{L} is convex if for all \mathcal{L} -TBoxes \mathcal{T} and \mathcal{L} -concepts C, D_1, \dots, D_n , $n \geq 2$ with $\mathcal{T} \models C \sqsubseteq D_1 \sqcup \dots \sqcup D_n$ we have $\mathcal{T} \models C \sqsubseteq D_i$ for some i .*

Note that, although disjunction \sqcup is not part of the syntax for $\text{Prob}\mathcal{EL}$, this is a valid definition: disjunction is just interpreted as in $\text{Prob}\mathcal{ALC}$. We will show that each single fragment $\text{Prob}\mathcal{EL}_c^{\sim p}$ is *non-convex* by providing a *non-convexity witness* which consists of a TBox \mathcal{T} and concepts C, D_1, \dots, D_n that violate the convexity property. Intuitively, non-convexity is used to reintroduce disjunction. This enables us to reduce concept satisfiability in \mathcal{ALC} relative to general TBoxes to subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p}$. We will divide the proof of Theorem 4.3 into four parts. We show:

- non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{<, \leq\}$ and $p \in (0, 1)$;
- non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{>, \geq, =\}$ and $p \leq 0.5$;
- non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{>, \geq, =\}$ and $p > 0.5$; and
- non-convexity implies EXPTIME-hardness.

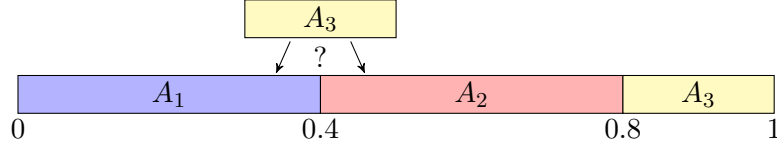


Figure 4.1: Intuition for non-convexity: A domain element satisfying $P_{\geq 0.4}A_1 \sqcap P_{\geq 0.4}A_2 \sqcap P_{\geq 0.4}A_3$ has to satisfy either $A_1 \sqcap A_2$, $A_1 \sqcap A_3$, or $A_2 \sqcap A_3$ in some world.

Non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{<, \leq\}$

For this case, choose some $\sim \in \{<, \leq\}$ and p in $(0, 1)$. There is a very simple argument for non-convexity relative even to the empty TBox as we have

$$\emptyset \models \top \sqsubseteq P_{\sim p}A \sqcup P_{\sim p}P_{\sim p}A \quad (4.1)$$

but both

$$\emptyset \not\models \top \sqsubseteq P_{\sim p}A \quad \text{and} \quad \emptyset \not\models \top \sqsubseteq P_{\sim p}P_{\sim p}A. \quad (4.2)$$

For proving (4.1), observe that for every individual $d \in \Delta^{\mathcal{I}}$, either $d \in (P_{\sim p}A)^{\mathcal{I}, w}$ or $d \notin (P_{\sim p}A)^{\mathcal{I}, w}$. In the latter case, the semantics implies that $d \notin (P_{\sim p}A)^{\mathcal{I}, v}$ for all $v \in W$, thus $d \in (P_{=0}P_{\sim p}A)^{\mathcal{I}, w}$ and also $d \in (P_{\sim p}P_{\sim p}A)^{\mathcal{I}, w}$. The statements in (4.2) are also direct consequences of the semantics.

Non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{>, \geq, =\}$

Let us start with a small example illustrating the idea underlying the proof for this case. Take $p = 0.5$, $\sim = \geq$, and assume the TBox

$$\begin{aligned} \mathcal{T} = \{ & A_1 \sqcap A_2 \sqsubseteq P_{\geq 0.4}B_1, \\ & A_1 \sqcap A_3 \sqsubseteq P_{\geq 0.4}B_2, \\ & A_2 \sqcap A_3 \sqsubseteq P_{\geq 0.4}B_3 \}. \end{aligned}$$

Assume an arbitrary model \mathcal{I} of \mathcal{T} and a domain element d satisfying the concept $P_{\geq 0.4}A_1 \sqcap P_{\geq 0.4}A_2 \sqcap P_{\geq 0.4}A_3$. It should be clear that there is a world with positive probability where d satisfies two among A_1, A_2, A_3 , but there is a choice which ones these are; see Figure 4.1. By the TBox, the choice is ‘marked’ using $P_{\geq 0.4}B_1$, $P_{\geq 0.4}B_2$, or $P_{\geq 0.4}B_3$.

Let us generalize the above construction and consider $\text{Prob}\mathcal{EL}_c^{\geq p}$ with $p \leq 0.5$. Choose some $k > 0$ such that $k \cdot p > 1$ and set:

$$\begin{aligned} \mathcal{T} &= \{A_i \sqcap A_j \sqsubseteq P_{\geq p}B_{ij} \mid 1 \leq i < j \leq k\} \\ C &= P_{\geq p}A_1 \sqcap \dots \sqcap P_{\geq p}A_k \\ D_{ij} &= P_{\geq p}B_{ij} \end{aligned}$$

We show that the above witnesses non-convexity. Similar to the small example above, the probabilities stipulated by C sum up to > 1 , thus some of the A_i have to overlap, but there is a choice as to which ones these are.

Lemma 4.6. $\mathcal{T} \models C \sqsubseteq \sqcup_{1 \leq i < j \leq k} D_{ij}$, but $\mathcal{T} \not\models C \sqsubseteq D_{ij}$ for $1 \leq i < j \leq k$.

Proof. For the former, let \mathcal{I} be a model of \mathcal{T} and $d \in C^{\mathcal{I},w}$. Since $d \in (P_{\geq p}A_i)^{\mathcal{I},w}$ for $1 \leq i \leq k$ and $k \cdot p > 1$, there is a world v with $d \in (A_i \sqcap A_j)^{\mathcal{I},v}$ for some i, j with $1 \leq i < j \leq k$. It follows that $d \in D_{ij}^{\mathcal{I},v}$, thus $d \in D_{ij}^{\mathcal{I},w}$.

For the latter, fix arbitrary i_0, j_0 with $1 \leq i_0 < j_0 \leq k$. We show $\mathcal{T} \not\models C \sqsubseteq D_{i_0j_0}$ by constructing a model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} with $\Delta^{\mathcal{I}} = \{d\}$ and $W = \{w_1, w_2\}$ such that $d \in C^{\mathcal{I},v}$ and $d \notin D_{i_0j_0}^{\mathcal{I},v}$ for any $v \in W$. Formally, we set for $1 \leq i \leq k$:

$$\begin{aligned} A_i^{\mathcal{I},w_1} &:= \begin{cases} \emptyset & \text{if } i = i_0 \\ \{d\} & \text{otherwise} \end{cases} \\ A_i^{\mathcal{I},w_2} &:= \begin{cases} \{d\} & \text{if } i = i_0 \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

and for all $1 \leq i < j \leq k$:

$$B_{ij}^{\mathcal{I},w_1} := B_{ij}^{\mathcal{I},w_2} := \begin{cases} \emptyset & \text{if } i_0 \in \{i, j\}; \\ \{d\} & \text{otherwise.} \end{cases}$$

Finally, set $\mu(w_1) = \mu(w_2) = 0.5$. It is easy to check that \mathcal{I} is a model of \mathcal{T} . Moreover, we have $p_d^{\mathcal{I}}(A_i) = 0.5 \geq p$, i.e., $d \in C^{\mathcal{I},w}$ for any $w \in W$, and there is no world w with $d \in D_{i_0j_0}^{\mathcal{I},w}$. ■

When we replace \geq with $=$ in \mathcal{T} , C , and the D_{ij} , the first part of the proof of Lemma 4.6 still goes through, without any modifications. For the model construction we need to slightly modify the above interpretation: we add a new world w_3 with $A_i^{\mathcal{I},w_3} := B_{ij}^{\mathcal{I},w_3} := \emptyset$ and update the probability distribution as follows:

$$\begin{aligned} \mu(w_1) &:= \mu(w_2) := p \\ \mu(w_3) &:= 1 - 2p \end{aligned}$$

When we replace \geq with $>$ and assume $p < 0.5$, the proof of Lemma 4.6 goes through without any modifications. However, the case $\sim = >$ and $p = 0.5$ requires a slightly different construction. Set

$$\begin{aligned} \mathcal{T} &= \{A_i \sqcap A_j \sqcap A_k \sqsubseteq P_{\geq p}B_{ijk} \mid 1 \leq i < j < k \leq 4\} \\ C &= P_{\geq p}A_1 \sqcap \dots \sqcap P_{\geq p}A_4 \\ D_{ijk} &= P_{\geq p}B_{ijk} \end{aligned}$$

4 Subjective Uncertainty in \mathcal{EL}

It is only slightly more complicated to show that the above witnesses non-convexity, similarly to the proof of Lemma 4.6. Let us first show

$$\mathcal{T} \models C \sqsubseteq D_{123} \sqcup D_{124} \sqcup D_{134} \sqcup D_{234}$$

using again the pigeon hole principle. Assume a model \mathcal{I} of \mathcal{T} and $d \in C^{\mathcal{I},w}$. Since $d \in (P_{>0.5}A_i)^{\mathcal{I},w}$ for $1 \leq i \leq 4$, there is a world v with $d \in (A_i \sqcap A_j \sqcap A_k)^{\mathcal{I},v}$ for some $1 \leq i < j < k \leq 4$. By the TBox, $d \in D_{ijk}^{\mathcal{I},v}$.

For the second part, we show only $\mathcal{T} \not\models C \sqsubseteq D_{123}$ as the other cases are symmetric. For this purpose, we define a model $\mathcal{I} = (\{d\}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} with $W = \{w_1, \dots, w_4\}$ such that $d \in C^{\mathcal{I},w_4}$ but $d \notin D_{123}^{\mathcal{I},w}$ for some $w \in W$.

- $A_1^{\mathcal{I},v} = \{d\}$ for $v \in \{w_1, w_2, w_3\}$;
- $A_2^{\mathcal{I},v} = \{d\}$ if $v \in \{w_1, w_4\}$;
- $A_3^{\mathcal{I},v} = \{d\}$ if $v \in \{w_2, w_4\}$;
- $A_4^{\mathcal{I},v} = \{d\}$ if $v \in \{w_3, w_4\}$;
- $B_{234}^{\mathcal{I},v} = \{d\}$ if $v \in \{w_1, w_2, w_3\}$;
- $D_{234}^{\mathcal{I},v} = \{d\}$ for all v ;
- the extension of any concept that is not mentioned is empty.

Finally, set $\mu(w_1) = \mu(w_2) = \mu(w_3) = 0.2$ and $\mu(w_4) = 0.4$. It is not hard to check that $\mathcal{I} \models \mathcal{T}$. Moreover, we have $p_d^{\mathcal{I}}(A_i) = 0.6 > 0.5$ for all $1 \leq i \leq 4$, i.e., $d \in C^{\mathcal{I},w}$ for all $w \in W$. On the other hand, there is no world $w \in W$ with $d \in D_{123}^{\mathcal{I},w}$.

Non-convexity of $\text{Prob}\mathcal{EL}_c^{\sim p}$ with $\sim \in \{>, \geq, =\}$ and $p > 0.5$

The main idea here is to use the constructor $P_{\sim p}C$ to simulate $P_{>q}C$, for some $q < 0.5$. First let $\sim = >$ and fix a $p > 0.5$. Let $n > 0$ be the smallest integer such that $n > \frac{1}{2(1-p)}$ and set $q = pn - n + 1$. Note that $n \geq 2$ and $0 \leq q < 0.5$. Intuitively, we use the fact that

$$P_{>p}X_1 \sqcap \dots \sqcap P_{>p}X_n \sqsubseteq P_{>q}(X_1 \sqcap \dots \sqcap X_n),$$

which allows us to simply redo the above reduction with probability $q < 0.5$. Thus, let $k > 0$ be such that $k \cdot q > 1$ and define

$$\begin{aligned} \mathcal{T} &= \{A_{i1} \sqcap \dots \sqcap A_{in} \sqsubseteq A_i \mid 1 \leq i \leq k\} \cup \\ &\quad \{A_i \sqcap A_j \sqsubseteq P_{>p}B_{ij} \mid 1 \leq i < j \leq k\} \\ C &= \bigcap_{1 \leq i \leq k} \bigcap_{1 \leq \ell \leq n} P_{>p}A_{i\ell} \\ D_{ij} &= P_{>p}B_{ij} \end{aligned}$$

Indeed, the above witnesses non-convexity.

Lemma 4.7. $\mathcal{T} \models C \sqsubseteq \sqcup_{1 \leq i < j \leq k} D_{ij}$, but $\mathcal{T} \not\models C \sqsubseteq D_{ij}$ for $1 \leq i < j \leq k$.

Proof. For the former, let \mathcal{I} be a model of \mathcal{T} and $d \in C^{\mathcal{I},w}$. We first verify the following:

Claim. $d \in (P_{>q}A_i)^{\mathcal{I},w}$ for all $1 \leq i \leq k$.

Proof of the Claim. Assume this is not the case for some i , i.e., $d \in (P_{\leq q}A_i)^{\mathcal{I},w}$. Define

$$S := \sum_{1 \leq \ell \leq n} p_d^{\mathcal{I}}(A_{i\ell}).$$

On the one hand, we must have $S > p \cdot n$ since $d \in C^{\mathcal{I},w}$. On the other hand, given an interpretation \mathcal{I} , we can compute S as follows:

$$S = \sum_{w \in W} \mu(w) \cdot |\{1 \leq \ell \leq n \mid d \in A_{i\ell}^{\mathcal{I},w}\}|. \quad (4.3)$$

Let us partition W into W_1, W_2 such that

- (i) $|\{1 \leq \ell \leq n \mid d \in A_{i\ell}^{\mathcal{I},w}\}| = n$ for all $w \in W_1$ and
- (ii) $|\{1 \leq \ell \leq n \mid d \in A_{i\ell}^{\mathcal{I},w}\}| < n$ for all $w \in W_2$.

By assumption, $d \in (P_{\leq q}A_i)^{\mathcal{I},w}$ and $A_{i1} \sqcap \dots \sqcap A_{in} \sqsubseteq A_i \in \mathcal{T}$. Thus, we obtain $p_d^{\mathcal{I}}(A_{i1} \sqcap \dots \sqcap A_{in}) \leq q$. Hence, we get $\mu(W_1) \leq q$. Together with Equation (4.3) this yields:

$$\begin{aligned} S &= \sum_{w \in W_1} \mu(w) \cdot |\{1 \leq \ell \leq n \mid d \in A_{i\ell}^{\mathcal{I},w}\}| + \sum_{w \in W_2} \mu(w) \cdot |\{1 \leq \ell \leq n \mid d \in A_{i\ell}^{\mathcal{I},w}\}| \\ &\leq n\mu(W_1) + (n-1)\mu(W_2) = n\mu(W_1) + (n-1)(1-\mu(W_1)) \\ &= n + \mu(W_1) - 1 \leq n + q - 1. \end{aligned}$$

It remains to apply $q = pn - n + 1$ to obtain $S \leq pn$. Thus, overall we got $pn < S \leq pn$ which is a clear contradiction and finishes the proof of the claim.

We can now continue as in the proof of Lemma 4.6: since $k \cdot q > 1$, there is a world $v \in W$ with $d \in (A_i \sqcap A_j)^{\mathcal{I},v}$ for some i, j with $1 \leq i < j \leq k$. It follows that $d \in (P_{>p}B_{ij})^{\mathcal{I},v}$, thus $d \in D_{ij}^{\mathcal{I},v}$.

For showing that none of the D_{ij} is implied by C , fix some i_0, j_0 with $1 \leq i_0 < j_0 \leq k$. We construct a model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} with $\Delta^{\mathcal{I}} = \{d\}$ and $W = \{w_{1\ell}, w_{2\ell} \mid$

$1 \leq \ell \leq n\}$ such that $d \in C^{\mathcal{I},v}$ and $d \notin D_{i_0 j_0}^{\mathcal{I},v}$ for any $v \in W$ by setting for all $1 \leq i \leq k$ and $1 \leq \ell, \ell' \leq n$:

$$\begin{aligned} A_i^{\mathcal{I}, w_{1\ell}} &:= \begin{cases} \{d\} & \text{if } i \neq i_0; \\ \emptyset & \text{otherwise} \end{cases} \\ A_i^{\mathcal{I}, w_{2\ell}} &:= \begin{cases} \{d\} & \text{if } i = i_0; \\ \emptyset & \text{otherwise} \end{cases} \\ A_{i\ell'}^{\mathcal{I}, w_{1\ell}} &:= \begin{cases} \{d\} & \text{if } i \neq i_0 \text{ or } \ell \neq \ell'; \\ \emptyset & \text{otherwise} \end{cases} \\ A_{i\ell'}^{\mathcal{I}, w_{2\ell}} &:= \begin{cases} \{d\} & \text{if } i = i_0 \text{ or } \ell \neq \ell'; \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

and for all $1 \leq i < j \leq k$, $1 \leq \ell \leq n$:

$$\begin{aligned} B_{ij}^{\mathcal{I}, w_{1\ell}} &:= B_{ij}^{\mathcal{I}, w_{2\ell}} := \begin{cases} \{d\} & \text{if } i_0 \notin \{i, j\} \\ \emptyset & \text{otherwise} \end{cases} \\ \mu(w_{1\ell}) &:= \mu(w_{2\ell}) := \frac{1}{2n} \end{aligned}$$

Note that for every $A_{i\ell}$ with $1 \leq i \leq k$ and $1 \leq \ell \leq n$, there is a $b \in \{1, 2\}$ such that d satisfies $A_{i\ell}$ in $w_{b\ell'}$ for all ℓ' and in all $w_{(3-b)\ell'}$ whenever $\ell \neq \ell'$. Using additionally the fact that $n > \frac{1}{2(1-p)}$ we obtain:

$$p_d^{\mathcal{I}}(A_{i\ell}) = \frac{1}{2} + (n-1) \cdot \frac{1}{2n} = 1 - \frac{1}{2n} > 1 - \frac{1}{2 \cdot \frac{1}{2(1-p)}} = 1 - \frac{2(1-p)}{2} = p.$$

Thus, we have $d \in C^{\mathcal{I},v}$ for any $v \in W$. Further observe that $d \in A_{i_0}^{\mathcal{I},v}$ iff v is of the form $w_{2\ell}$ iff $d \notin A_{j_0}^{\mathcal{I},v}$. Thus, $d \notin D_{i_0 j_0}^{\mathcal{I},v}$ for any $v \in W$. It remains to verify that \mathcal{I} is a model of \mathcal{T} .

- Let $v \in W$ such that $d \in (A_{i_1} \sqcap \dots \sqcap A_{i_n})^{\mathcal{I},v}$ for some i . If $i \neq i_0$, v is of the form $w_{1\ell}$, by construction of \mathcal{I} . But then also $d \in A_i^{\mathcal{I},v}$. If $i = i_0$, then v is of the form $w_{2\ell}$ and $d \in A_{i_0}^{\mathcal{I},v}$.
- Let $v \in W$ such that $d \in (A_i \sqcap A_j)^{\mathcal{I},v}$ for some $i < j$. By construction, we have $i_0 \notin \{i, j\}$. Hence, $d \in (P_{=1} B_{ij})^{\mathcal{I},v}$.

This finishes the proof of the Lemma. ■

For the case $\sim = \geq$, we can use exactly the same construction and the proof goes through with only slight modifications. In case of equality, the first part of the proof goes through, but we have to change the model construction. We add a world w_3 such that in this

world d is not in the extension of any concept, that is, $A_{ij}^{\mathcal{I}, w_3} := A_i^{\mathcal{I}, w_3} := B_{ij}^{\mathcal{I}, w_3} := \emptyset$. Moreover, we need to modify the probability distribution μ in the following way:

$$\begin{aligned}\mu(w_{1\ell}) &:= \mu(w_{2\ell}) := \frac{p}{2n-1}; \\ \mu(w_3) &:= 1 - \frac{4np}{2n-1}.\end{aligned}$$

It is not hard to verify that $\mu(w_3) \geq 0$ since $p > 0.5$, $n > \frac{1}{2(p-1)}$, and $\sum_{w \in W} \mu(w) = 1$, thus μ is a valid probability distribution. Further, we can verify with the same arguments as in the proof above that $p_d^{\mathcal{I}}(A_{i\ell}) = p$, thus $d \in C^{\mathcal{I}, v}$ for every v . Finally, it is easy to check that $d \notin D_{i_0 j_0}^{\mathcal{I}, v}$ for any $v \in W$.

Non-Convexity implies EXPTIME-hardness

We apply a standard proof technique from [9] to show that non-convexity of \mathcal{EL} -variants implies EXPTIME-hardness; in particular, we use non-convexity to reduce from concept satisfiability relative to general \mathcal{ALC} -TBoxes which is known to be EXPTIME-complete. Fix any logic $\text{Prob}\mathcal{EL}_c^{\sim p}$.

Suppose that an \mathcal{ALC} -TBox \mathcal{T} and a concept name A_0 are given for which satisfiability is to be decided. It is well-known that we can without loss of generality assume that:

- the abbreviations for disjunction \sqcup and universal quantification $\forall r.C$ do not appear in the TBox;
- negation \neg occurs only in front of concept names (otherwise: introduce a fresh concept name A for every subconcept $\neg C$ in \mathcal{T} with C complex, replace $\neg C$ with $\neg A$, and add $A \sqsubseteq C$ and $C \sqsubseteq A$ to \mathcal{T}).

We eliminate negation as follows.

- (1) For every subconcept $\neg A$, introduce a fresh concept name \bar{A} , replace every occurrence of $\neg A$ with \bar{A} , and add $\top \sqsubseteq A \sqcup \bar{A}$ and $A \sqcap \bar{A} \sqsubseteq \perp$ to \mathcal{T} .
- (2) Eliminate the disjunctions $\top \sqsubseteq A \sqcup \bar{A}$ introduced in step (1): they are replaced with

$$\top \sqsubseteq \widehat{C}, \quad \widehat{D}_1 \sqsubseteq A, \quad \widehat{D}_i \sqsubseteq \bar{A} \quad \text{for } 2 \leq i \leq k,$$

and the concept inclusions from $\widehat{\mathcal{T}}$ where $\widehat{\mathcal{T}}, \widehat{C}, \widehat{D}_1, \dots, \widehat{D}_k$ is a fresh copy of the non-convexity witness that exists for $\text{Prob}\mathcal{EL}_c^{\sim p}$.

Let \mathcal{T}' be the TBox obtained by these manipulations. It is standard to prove that A_0 is satisfiable w.r.t. \mathcal{T} iff A_0 is satisfiable w.r.t. \mathcal{T}' . Note that \mathcal{T}' is a $\text{Prob}\mathcal{ALC}$ -TBox containing only the operators $\sqcap, \sqcup, \exists, \top, \perp$, and $P_{\sim p}$. Thus, it remains to deal with the concept \perp being the last one not allowed by our syntax for $\text{Prob}\mathcal{EL}_c^{\sim p}$. In particular, we reduce satisfiability of A_0 w.r.t. \mathcal{T}' to (non-)subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p}$. Introduce a

fresh concept name L and obtain \mathcal{T}'' from \mathcal{T}' by replacing every occurrence of \perp with L and adding the concept inclusions $\exists r.L \sqsubseteq L$ for every role name r from \mathcal{T}' . Then A_0 is satisfiable w.r.t. \mathcal{T}' iff $\mathcal{T}'' \not\models A_0 \sqsubseteq L$. This finishes the reduction and shows EXPTIME-hardness of $\text{Prob}\mathcal{EL}_c^{\sim p}$.

4.2.2 Subsumption Relative to Classical TBoxes

As we have seen in the previous section, checking subsumption relative to general $\text{Prob}\mathcal{EL}_c^{\sim p}$ -TBoxes is EXPTIME-hard for all $\sim \in \{<, \leq, =, >, \geq\}$ and $p \in (0, 1)$. It has been shown in [101] that for $p \in \{0, 1\}$, PTIME can be reattained. The goal in this part is to study another possibility to stay in PTIME, namely by the restriction to *classical* TBoxes.

Theorem 4.8. *For all $\sim \in \{>, \geq\}$ and $p \in [0, 1]$, subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p;=1}$ relative to classical TBoxes is in PTIME.*

Observe that the theorem is a corollary of the PTIME result for subsumption in $\text{Prob}\mathcal{EL}_c^{>0;=1}$ relative to general TBoxes in case $p = 0$ and $\sim = >$ [101]. For proving the theorem for the remaining cases, we need some notation. A concept name A is *defined* in a classical TBox \mathcal{T} if there is a concept definition $A \equiv C \in \mathcal{T}$, and *primitive* otherwise. For a given TBox \mathcal{T} and a defined concept name A in \mathcal{T} , we refer with C_A to the *defining concept* for A in \mathcal{T} , i.e., $A \equiv C_A \in \mathcal{T}$. Moreover, we deliberately confuse the concept $C_A = D_1 \sqcap \dots \sqcap D_k$ with the set $\{D_1, \dots, D_k\}$. We can without loss of generality restrict our attention to the subsumption of defined concept names and, moreover, assume that the input TBox is normalized to a set of concept definitions of the form

$$A \equiv P_1 \sqcap \dots \sqcap P_n \sqcap C_1 \sqcap \dots \sqcap C_m,$$

where $n, m \geq 0$, and P_1, \dots, P_n are primitive concept names, and C_1, \dots, C_m are of the form $P_{\sim p}A$, $P_{=1}A$, or $\exists r.A$ with A a defined concept name (note that the top concept is completely normalized away). It is well-known that such a normalization can be achieved in polynomial time, see for instance [8] for details. Finally, we define a set of concepts *certain* for C_A as

$$\text{cert}(C_A) = \{P_*B \mid P_*B \in C_A\} \cup \bigcup_{P_{=1}B \in C_A} \{C_B\},$$

where, here and in what follows, P_* ranges over $P_{=1}$ and $P_{\sim p}$. Intuitively, $\text{cert}(C_A)$ contains concepts that hold with probability 1 whenever A is satisfied in some world.

Our main tool for proving the PTIME upper bound is a consequence-driven algorithm similar to the ones in [9, 89]. The algorithm is depicted in Figure 4.2. The algorithm starts with a normalized input TBox and then exhaustively applies the completion rules displayed in Figure 4.2. As a general proviso, each rule can be applied only if it adds a

R1	If $\exists r.B \in C_A$, and $C_{B'} \subseteq C_B$, then replace $A \equiv C_A$ with $A \equiv C_A \cup \{\exists r.B'\}$
R2	If $P_{=1}B \in C_A$, then replace $A \equiv C_A$ with $A \equiv C_A \cup C_B$
R3	If $P_{=1}B \in C_A$, then replace $A \equiv C_A$ with $A \equiv C_A \cup \{P_{\sim p}B\}$
R4	If $P_{\sim p}B \in C_A$, and $D \in \text{cert}(C_B)$, then replace $A \equiv C_A$ with $A \equiv C_A \cup \{D\}$
R5	If $C_B \subseteq \text{cert}(C_A)$, then replace $A \equiv C_A$ with $A \equiv C_A \cup \{P_{=1}B\}$
R6	If $P_{\sim p}B \in C_A$ and $C_{B'} \subseteq \text{cert}(C_A) \cup C_B$, then replace $A \equiv C_A$ with $A \equiv C_A \cup \{P_{\sim p}B'\}$

Figure 4.2: TBox completion rules for subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p,=1}$.

concept that occurs in \mathcal{T} and actually changes the TBox, e.g., **R1** can only be applied when $\exists r.B'$ occurs in \mathcal{T} and $\exists r.B' \notin C_A$. Exemplarily, we explain rule **R5** in more detail. If all defining concepts C_B of B are contained in the certain concepts for A , then we can add $P_{=1}B$ to C_A since in this case we have $\mathcal{T} \models A \sqsubseteq P_{=1}B$. In the following lemma, we establish correctness of the algorithm. The “only if” direction requires a careful and subtle model construction.

Lemma 4.9. *For all defined concept names A, B , we have $\mathcal{T} \models A \sqsubseteq B$ iff, after exhaustive rule application, $C_B \subseteq C_A$.*

Proof. We start with soundness, that is, the “if”-direction. Using the semantics, it is straightforward to show that the rules are correct, i.e., if a TBox \mathcal{T}_2 is obtained from a TBox \mathcal{T}_1 by a single rule application, then every model of \mathcal{T}_1 is also a model of \mathcal{T}_2 (and clearly, vice versa). More precisely, let $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ be any model of \mathcal{T}_1 and assume $d \in \Delta^{\mathcal{I}}$ and $w \in W$ such that $d \in A^{\mathcal{I},w}$ and hence $d \in (C_A)^{\mathcal{I},w}$. We show that if a concept D is added to C_A , then we also have $d \in D^{\mathcal{I},w}$.

R1 Assume $\exists r.B \in C_A$ and $C_{B'} \subseteq C_B$. By the former, we have $d \in (\exists r.B)^{\mathcal{I},w}$, i.e., there is some element $e \in \Delta^{\mathcal{I}}$ with $e \in B^{\mathcal{I},w}$ and $(d, e) \in r^{\mathcal{I},w}$. As \mathcal{I} is a model of \mathcal{T}_1 , we also have $e \in (C_B)^{\mathcal{I},w}$ and, by the assumption $C_{B'} \subseteq C_B$, we obtain $e \in (C_{B'})^{\mathcal{I},w}$. Again, as $\mathcal{I} \models \mathcal{T}_1$, we have $e \in B'^{\mathcal{I},w}$. The semantics yields $d \in (\exists r.B')^{\mathcal{I},w}$.

- R2** Assume $P_{=1}B \in C_A$. Hence, we have $d \in (P_{=1}B)^{\mathcal{I},w}$. By the semantics, $d \in B^{\mathcal{I},w}$ and thus $d \in (C_B)^{\mathcal{I},w}$.
- R3** Assume $P_{=1}B \in C_A$. Thus, we have $d \in (P_{=1}B)^{\mathcal{I},w}$ which implies $d \in (P_{\sim p}B)^{\mathcal{I},w}$.
- R4** Assume $P_{\sim p}B \in C_A$ and $D \in \text{cert}C_B$. By the former, we have $d \in (P_{\sim p}B)^{\mathcal{I},w}$. Hence, there is some world $v \in W$ with $d \in B^{\mathcal{I},v}$. By definition of $\text{cert}(C_B)$, we have that $d \in E^{\mathcal{I},w'}$ for all $E \in \text{cert}(C_B)$ and $w' \in W$. Thus, in particular, $d \in D^{\mathcal{I},w}$.
- R5** Assume $C_B \subseteq \text{cert}(C_A)$. By definition of $\text{cert}(C_A)$, we have that $d \in E^{\mathcal{I},v}$ for all $E \in \text{cert}(C_A)$ and $v \in W$. As $C_B \subseteq \text{cert}(C_A)$, we have $d \in (C_B)^{\mathcal{I},v}$ for all $v \in W$. As \mathcal{I} is a model of \mathcal{T}_1 , we get $d \in B^{\mathcal{I},v}$ for all $v \in W$, thus $d \in (P_{=1}B)^{\mathcal{I},w}$.
- R6** Assume $P_{\sim p}B \in C_A$ and $C_{B'} \subseteq \text{cert}(C_A) \cup C_B$. By the former, we have $d \in (P_{\sim p}B)^{\mathcal{I},w}$ and hence $p_d^{\mathcal{I}}(C_B) \sim p$. By definition of $\text{cert}(C_A)$ and the semantics, we have $p_d^{\mathcal{I}}(\text{cert}(C_A)) = 1$. Together, this implies $p_d^{\mathcal{I}}(\text{cert}(C_A) \sqcap C_B) \sim p$. As by assumption $C_{B'} \subseteq C_A \cup C_B$, we also have $p_d^{\mathcal{I}}(C_{B'}) \sim p$ and hence $d \in (P_{\sim p}B')^{\mathcal{I},w}$.

To finish this direction, assume an arbitrary model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ of \mathcal{T} and some $d \in \Delta^{\mathcal{I}}$ and $w \in W$ such that $d \in A^{\mathcal{I},w}$. We showed that \mathcal{I} is also a model of the TBox obtained after exhaustive rule application. Let C_A, C_B the definitions of A, B after the algorithm stopped. By the semantics, we have $d \in (C_A)^{\mathcal{I},w}$. The assumption $C_B \subseteq C_A$ yields $d \in (C_B)^{\mathcal{I},w}$. Applying the semantics again implies $d \in B^{\mathcal{I},w}$.

For “only if”, let $C_{B_0} \not\subseteq C_{A_0}$ for some defined concept names A_0, B_0 and assume $\sim = >$. Our aim is to construct a model \mathcal{I} of \mathcal{T} that witnesses $\mathcal{T} \not\models A_0 \sqsubseteq B_0$. Let Def denote the set of defined concept names in \mathcal{T} . Moreover, we use BC to denote all *basic concepts*, that is, concepts of the form P (primitive concept name), $\exists r.A$, $P_{=1}A$, and $P_{\sim p}A$ that occur in \mathcal{T} . We first fix some constants that will be used to define the probabilities of worlds in the model \mathcal{I} :

- first fix $\alpha, \alpha' \in (0, 1)$ such that $\frac{\alpha}{2} < \alpha' < \alpha < p$ (possible since $p > 0$);
- next fix an integer $m \geq 2$ such that

$$(p - \alpha') + \frac{1 - (p - \alpha' + 3|\text{Def}| \cdot \frac{\alpha}{2})}{m} < p$$

(this can be done simply by choosing m sufficiently large as $p - \alpha' < p$);

- finally, we choose an integer $k \geq 2$ such that

$$(p - \alpha') \cdot \frac{k-1}{k} + \alpha > p$$

(this can be done again by choosing k sufficiently large since $(p - \alpha') + \alpha > p$).

Start with defining an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ by setting:

$$\begin{aligned} W &= \{\delta_{Ai}, 1_j, p_\ell \mid A \in \text{Def}, i \in \{1, 2, 3\}, 1 \leq j \leq m, 1 \leq \ell \leq k\} \\ \Delta^{\mathcal{I}} &= \{(A, v) \mid A \in \text{Def}, v \in W\} \\ \mu(p_\ell) &= \frac{p - \alpha'}{k} \\ \mu(\delta_{Ai}) &= \frac{\alpha}{2} \\ \mu(1_j) &= \frac{1 - (p - \alpha' + 3|\text{Def}| \cdot \frac{\alpha}{2})}{m} \end{aligned}$$

It is readily checked that the sum $\sum_{w \in W} \mu(w)$ adds up to 1 as required. Moreover, observe the following two important properties of μ :

- (P1) for any set V of worlds that contains at least $k - 1$ of the worlds in $\{p_\ell \mid \ell \leq k\}$ and at least two distinct δ_{Ai}, δ_{Bj} the probabilities sum up to more than p ;
- (P2) any set of worlds whose probabilities sum up to a value $> p$ includes at least two worlds from $W \setminus \{p_\ell \mid \ell \leq k\}$.

Using the condition in the choice of k it is not hard to see that Property (P1) is satisfied. For Property (P2), define $V = \{p_\ell \mid \ell \leq k\}$ and observe that $\mu(V) = p - \alpha'$. Now, by the choice of α and α' it is clear that $\mu(V \cup \{\delta_{Ai}\}) = p - \alpha' + \frac{\alpha}{2} < p$. Finally, by the choice of m , we have that $\mu(V \cup \{1_j\}) = p - \alpha' + (1 - (p - \alpha' + 3|\text{Def}| \cdot \frac{\alpha}{2}))/m < p$.

To define concept and role memberships, first define a map $\pi : (\Delta^{\mathcal{I}} \times W) \rightarrow 2^{\text{BC}}$ such that each set $\pi(\cdot, \cdot)$ is minimal with the following conditions satisfied for all $A \in \text{Def}$ and $v, w \in W$:

1. $C_A \subseteq \pi((A, w), w)$;
2. if $P_*B \in C_A$, then $P_*B \in \pi((A, w), v)$;
3. if $P_{=1}B \in C_A$, then $C_B \subseteq \pi((A, w), v)$;
4. if $P_{>p}B \in C_A$, then $C_B \subseteq \pi((A, w), p_\ell)$ for all $\ell \leq k$ when $w \notin \{p_\ell \mid \ell \leq k\}$;
5. if $P_{>p}B \in C_A$, then $C_B \subseteq \pi((A, p_i), p_\ell)$ for all $\ell \leq k$ with $i \neq \ell$;
6. if $P_{>p}B \in C_A$, then $C_B \subseteq \pi((A, w), \delta_{B1})$ and $C_B \subseteq \pi((A, w), \delta_{B2})$ when $w \notin \{\delta_{B1}, \delta_{B2}, \delta_{B3}\}$;
7. if $P_{>p}B \in C_A$, then $C_B \subseteq \pi((A, \delta_{Bi}), \delta_{Bj})$ for all distinct $i, j \in \{1, 2, 3\}$;

Now define the interpretation of the defined concept names A , primitive concept names P , and role names r as

$$\begin{aligned} A^{\mathcal{I}, w} &= \{d \in \Delta^{\mathcal{I}} \mid C_A \subseteq \pi(d, w)\}; \\ P^{\mathcal{I}, w} &= \{d \in \Delta^{\mathcal{I}} \mid P \in \pi(d, w)\}; \\ r^{\mathcal{I}, w} &= \{(d, (B, w)) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists r.B \in \pi(d, w)\}. \end{aligned}$$

We establish the following, central claim.

Claim. For all $D \in \mathbf{BC}$, $(A, w) \in \Delta^{\mathcal{I}}$, and $v \in W$, we have:

$$(A, w) \in D^{\mathcal{I}, v} \quad \text{iff} \quad D \in \pi((A, w), v).$$

Proof of claim. We start with the “if” direction. Let $D \in \pi((A, w), v)$. We distinguish the following cases:

- $D = P$ is a primitive concept name. Immediate by definition of $P^{\mathcal{I}}$.
- $D = P_{=1}B$. Since $D \in \pi((A, w), v)$, the definition of π implies that one of the following cases applies:
 - If $D \in \pi((A, w), v)$ because of Condition 1 or 2, then $B \in \pi((A, w), w')$ for all w' . By definition of $B^{\mathcal{I}}$, it follows that $(A, w) \in (P_{=1}B)^{\mathcal{I}, w}$ as required.
 - Assume $D \in \pi((A, w), v)$ because of Condition 3, that is, $P_{=1}B' \in C_A$ and $D \in C_{B'}$. Due to rule **R2**, we have $D \in C_A$ and can argue as in the previous case.
 - In all other cases, we have $P_{>p}B' \in C_A$, and $D \in C_{B'}$. Thus, the definition of $\text{cert}(\cdot)$ yields $D \in \text{cert}(C_{B'})$. By **R4**, we have $D \in C_A$ and can again argue as before.
- $D = P_{>p}B$. We distinguish the same cases as above, i.e.,
 - $D \in C_A$. Then $C_B \subseteq \pi((A, w), p)$ for at least $k - 1$ distinct worlds p from $\{p_\ell \mid \ell \leq k\}$ (by Conditions 4 and 5 above) and $C_B \subseteq \pi((A, w), \delta_{Bi})$ and $C_B \subseteq \pi((A, w), \delta_{Bj})$ for distinct $i, j \in \{1, 2, 3\}$ (Conditions 6 and 7). By (P1) and definition of $B^{\mathcal{I}}$, it follows that $(A, w) \in (P_{>p}B)^{\mathcal{I}, v}$, as required.
 - $P_{=1}B' \in C_A$ and $D \in C_{B'}$. By rule **R2**, we have $D \in C_A$ and can argue as in the previous case.
 - $P_{>p}B' \in C_A$, and $D \in C_{B'}$. By definition of $\text{cert}(\cdot)$, we have $D \in \text{cert}(C_{B'})$. By **R4**, we have $D \in C_A$ and can again argue as before.
- $D = \exists r.B$. By definition of $r^{\mathcal{I}}$, we have $((A, w), (B, v)) \in r^{\mathcal{I}, v}$. By Condition 1 of π and definition of $B^{\mathcal{I}}$, we have $(B, v) \in B^{\mathcal{I}, v}$. Finally, the semantics yields $(A, w) \in (\exists r.B)^{\mathcal{I}, v}$.

For the “only if” direction, assume that $(A, w) \in D^{\mathcal{I}, v}$. Distinguish the following cases:

- $D = P$ is a primitive concept name. Immediate by definition of $P^{\mathcal{I}}$.
- $D = P_{=1}B$. Take a $1_j \in W$ such that $w \neq 1_j$ (exists since there are at least two worlds of the form 1_j). Since $(A, w) \in D^{\mathcal{I}, v}$, we have $(A, w) \in B^{\mathcal{I}, 1_j}$. By definition of $B^{\mathcal{I}}$, we thus have $C_B \subseteq \pi((A, w), 1_j)$. By definition of $\pi((A, w), 1_j)$, it follows that for every $D' \in C_B$, we have

- (i) $D' \in C_A$ with D' of the form P_*B' (by Condition 2 of π), or
- (ii) there is a $P_{=1}B' \in C_A$ with $D' \in C_{B'}$ (by Condition 3).

Thus, $C_B \subseteq \text{cert}(C_A)$ and rule **R5** yields $P_{=1}B \in C_A$. By Condition 2 of π , we have $P_{=1}B \in \pi((A, w), v)$ as required.

- $D = P_{>p}B$. Since $(A, w) \in (P_{>p}B)^{\mathcal{I}, v}$, (P2) yields the following cases:
 - $(A, w) \in B^{\mathcal{I}, 1_j}$ for some j with $w \neq 1_j$. Then we can argue as in the previous case that $P_{=1}B \in C_A$. Thus rule **R3** yields $P_{>p}B \in C_A$ and by Condition 2 of π , we have $P_{>p}B \in \pi((A, w), v)$ as required.
 - $(A, w) \in B^{\mathcal{I}, \delta_{B'j}}$ for some j with $w \neq \delta_{B'j}$. By definition of $B^{\mathcal{I}}$, we thus have $C_B \subseteq \pi((A, w), \delta_{B'j})$. By definition of $\pi((A, w), \delta_{B'j})$, it follows that for every $D' \in C_B$, we have
 - (i) $D' \in C_A$ with D' of the form P_*B'' (Condition 2 of π),
 - (ii) there is a $P_{=1}B'' \in C_A$ with $D' \in C_{B''}$ (Condition 3), or
 - (iii) $P_{>p}B' \in C_A$ and $D' \in C_{B'}$ (Condition 6 or 7).

If exclusively (i) and (ii) apply, then $C_B \subseteq \text{cert}(C_A)$; otherwise, we have $P_{>p}B' \in C_A$ and $C_B \subseteq \text{cert}(C_A) \cup C_{B'}$. In the first case, **R5** yields $P_{=1}B \in C_A$ and **R3** yields $P_{>p}B \in C_A$. In the latter case, **R6** yields $P_{>p}B \in C_A$. By Condition 2 of π , we have $P_{>p}B \in \pi((A, w), v)$ as required.

- $D = \exists r.B$. Then there is an (A', v) such that $((A, w), (A', v)) \in r^{\mathcal{I}, v}$ and $(A', v) \in B^{\mathcal{I}, v}$. By definition of $r^{\mathcal{I}, v}$, we have $\exists r.A' \in \pi((A, w), v)$. By definition of $B^{\mathcal{I}}$, we have $C_B \subseteq \pi((A', v), v)$. By definition of π , Conditions 4 to 7 cannot be responsible since we are concerned here with a set $\pi((A, v), v')$ with $v = v'$. By Conditions 1-3 of π , it follows that for every $D' \in C_B$ we have $D' \in C_{A'}$ or $P_{=1}B' \in C_{A'}$ with $D' \in C_{B'}$. In the latter case, we also obtain $B \in C_A$ by **R2**. Thus $C_B \subseteq C_{A'}$. To continue, we make a case distinction as follows:
 - $v = w$. Then the definition of π yields that $\exists r.A' \in C_A$ or $P_{=1}B' \in C_A$ with $\exists r.A' \in C_{B'}$. In the latter case, we also obtain $\exists r.A' \in C_A$ by **R2**. This, $C_B \subseteq C_{A'}$, and **R1** yield $\exists r.B \in C_A$. By definition of π , we thus have $\exists r.B \in \pi((A, w), v)$.
 - $v = 1_j, v \neq w$. Since $\exists r.A' \in \pi((A, w), v)$, the definition of π yields a $P_{=1}B' \in C_A$ with $\exists r.A' \in C_{B'}$. By **R1** and $C_B \subseteq C_{A'}$, we have $\exists r.B \in C_{B'}$. Thus, Condition 3 of π yields $\exists r.B \in \pi((A, w), v)$ as required.
 - $v = p_\ell, v \neq w$. Since $\exists r.A' \in \pi((A, w), v)$, the definition of π implies that there is a $P_{=1}B' \in C_A$ with $\exists r.A' \in C_{B'}$ (Condition 3) or a $P_{>p}B' \in C_A$ with $\exists r.A' \in C_{B'}$ (Conditions 4 and 5). In the former case, we can continue as in

the case $v = 1_j$ above. In the latter case, **R1** and $C_B \subseteq C_{A'}$ yield $\exists r.B \in C_{B'}$. Thus, Conditions 4 and 5 of π yield $\exists r.B \in \pi((A, w), v)$ as required.

- $v = \delta_{Ej}$, $v \neq w$. The reasoning is the same as in the previous case; we give it here for the sake of completeness. Since $\exists r.A' \in \pi((A, w), v)$, the definition of π implies that there is a $P_{=1}B' \in C_A$ with $\exists r.A' \in C_{B'}$ (Condition 3) or a $P_{>p}B' \in C_A$ with $\exists r.A' \in C_{B'}$ (Condition 6 and 7). In the former case, we can continue as in the case $v = 1_j$. In the latter case, **R1** and $C_B \subseteq C_{A'}$ yield $\exists r.B \in C_{B'}$. Thus, Conditions 6 and 7 of π yield $\exists r.B \in \pi((A, w), v)$ as required.

This finishes the proof of the claim.

It is an immediate consequence of the Claim and the interpretation of defined concept names that \mathcal{I} is a model of the TBox obtained after exhaustive rule applications. As this is equivalent to the input TBox (see “if”-direction) it is also a model of \mathcal{T} .

By Condition 1 on π , definition of $A_0^{\mathcal{I}}$, and the Claim, we have $(A_0, 1_1) \in A_0^{\mathcal{I}, 1_1}$. It thus remains to show that $(A_0, 1_1) \notin B_0^{\mathcal{I}, 1_1}$. Assume that the contrary holds. By definition of $B_0^{\mathcal{I}, 1_1}$, this means that $C_{B_0} \subseteq \pi((A_0, 1_1), 1_1)$. By definition of π , it follows that for every $D \in C_{B_0}$, we have $D \in C_{A_0}$ or $P_{=1}B \in C_{A_0}$ and $D \in C_B$. In the latter case, however, **R2** yields $D \in C_{A_0}$. In summary, we thus have $C_{B_0} \subseteq C_{A_0}$, which is a contradiction to our initial assumption that $C_{B_0} \not\subseteq C_{A_0}$.

This finishes the proof in case $\sim = >$. It turns out that when $\sim = \geq$, we can use exactly the same interpretation \mathcal{I} . The reason is that the definition of μ also satisfies the following variations of (P1) and (P2):

- (P1') for any set V of worlds that contains at least $k - 1$ of the worlds in $\{p_\ell \mid \ell \leq k\}$ and at least two distinct $\delta_{A_i}, \delta_{B_j}$ the probabilities sum up to at least p ;
- (P2') any set of worlds whose probabilities sum up to a value $\geq p$ includes at least two worlds from $W \setminus \{p_\ell \mid \ell \leq k\}$.

The rest of the proof is then identical. ■

It remains to verify that the completion algorithm in Figure 4.2 requires only polynomial time. It is clear that the precondition of every rule can be checked in polynomial time. Further, every successful rule application extends the TBox, and both the number of concept definitions and of conjuncts in each concept definition is bounded by the size of the original TBox. This finishes the proof of Theorem 4.8.

It is interesting to note that the proof of Theorem 4.8 is based on exactly the same algorithm, for all $\sim \in \{\geq, >\}$ and $p \in (0, 1)$. It follows that there is in fact only *one single logic* $\text{Prob}\mathcal{EL}_c^{\sim p}$, for all such \sim and p . Formally, given a $\text{Prob}\mathcal{EL}_c^{\sim p}$ -concept C , $\approx \in \{\geq, >\}$ and $q \in (0, 1]$, let $C_{\approx q}$ denote the result of replacing each subconcept $P_{\sim p}D$ in C with $P_{\approx q}D$ in C and similarly for $\text{Prob}\mathcal{EL}_c^{\sim p}$ -TBoxes \mathcal{T} .

Corollary 4.10. *For any $p, q > 0$, $\sim, \approx \in \{>, \geq\}$, $\text{Prob}\mathcal{EL}_c^{\sim p}$ -concepts C, D and $\text{TBox } \mathcal{T}$, we have $\mathcal{T} \models C \sqsubseteq D$ iff $\mathcal{T}_{\approx q} \models C_{\approx q} \sqsubseteq D_{\approx q}$.*

Consequently, the (potentially difficult!) choice of a concrete $\sim \in \{\geq, >\}$ and $p \in (0, 1]$ is moot. In fact, it might be more intuitive to replace the constructor $P_{\sim p}C$ with a constructor $\mathcal{L}C$ that describes elements which ‘are likely to be a C ’, and to replace $P_{=1}C$ with the constructor $\mathcal{C}C$ to describe elements that ‘are certain to be a C ’, see e.g. [70, 73] for other approaches to logics of likelihood. Note that the case $p = 0$ is different from the cases considered above: for example, we have $\mathcal{T}_\emptyset \models \exists r.A \sqsubseteq \exists r.P_{>p}A$ iff $p = 0$, and likewise $\mathcal{T}_\emptyset \models P_{>p}\exists r.A \sqsubseteq P_{>p}\exists r.P_{>p}A$ iff $p = 0$. In the spirit of the constructors \mathcal{C} and \mathcal{L} , $P_{>0}C$ can be replaced with a constructor $\mathcal{P}C$ that describes elements for which ‘it is possible that they are a C ’. For example, the SNOMED CT concepts *definite thrombus* and *possible thrombus* can then be written as $\mathcal{C}\text{Thrombus}$ and $\mathcal{P}\text{Thrombus}$.

As a consequence of the proof of Theorem 4.8 we obtain that the logics $\text{Prob}\mathcal{EL}_c^{\sim p;=1}$ are convex.

Corollary 4.11. *The logics $\text{Prob}\mathcal{EL}_c^{\sim p;=1}$ are convex relative to classical TBoxes for $\sim \in \{>, \geq\}$ and $p \in (0, 1)$.*

Proof. We assume $\mathcal{T} \not\models C \sqsubseteq D_1$ and $\mathcal{T} \not\models C \sqsubseteq D_2$. An inspection of the proof of the completeness part of Lemma 4.9 yields the existence of a probabilistic interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ such that, for every defined concept name $A \in \text{Def}$, there is a domain element $d_A \in \Delta^{\mathcal{I}}$ and a world $w_A \in W$ with $d_A \in A^{\mathcal{I}, w_A}$ and for all defined concept names $B \in \text{Def}$ we have $d_A \in B^{\mathcal{I}, w_A}$ if and only if $\mathcal{T} \models A \sqsubseteq B$. In this particular case, the assumption yields $d_C \in C^{\mathcal{I}, w_C}$, but $d_C \notin D_1^{\mathcal{I}, w_C}$ and $d_C \notin D_2^{\mathcal{I}, w_C}$. Thus, we clearly have $d_C \notin (D_1 \sqcup D_2)^{\mathcal{I}, w_C}$ which witnesses $\mathcal{T} \not\models C \sqsubseteq D_1 \sqcup D_2$. ■

It is natural to ask what happens in case of the other comparison operators $=$, $<$, and \leq . In case of $=$, we believe that the procedure in Figure 4.2 also applies to $\text{Prob}\mathcal{EL}_c^{\leq p}$: obviously, the rules remain sound and we believe that the ideas underlying the model construction in the completeness proof can be adapted. In case of the $<$ and \leq , the previous section about general TBoxes already suggested that these operators are somewhat more difficult as for each such logic non-convexity is witnessed already using an *empty* TBox \emptyset . Indeed one can prove the following theorem (not mentioning a TBox!).

Theorem 4.12. *Let $\sim \in \{<, \leq\}$ and $p \in (0, 1)$. Then, checking subsumption between $\text{Prob}\mathcal{EL}_c^{\sim p}$ -concepts is coNP-hard.*

Proof. The proof is by a reduction from UNSAT, i.e., checking whether a given propositional formulas in conjunctive normal form (CNF) is unsatisfiable. The proof is uniform for every $p \in (0, 1)$ and $\sim \in \{<, \leq\}$, hence we fix some p and consider $\text{Prob}\mathcal{EL}_c^{\leq p}$.

Let $\varphi = \varphi_1 \wedge \dots \wedge \varphi_n$ be a propositional formula in CNF with $\varphi_i = \ell_{i1} \vee \dots \vee \ell_{ik_i}$ where each literal ℓ_{ij} is either a variable x or a negated variable $\neg x$. Assume moreover that the set of occurring variables is $\{x_1, \dots, x_m\}$ and introduce corresponding concept names X_1, \dots, X_m . Define the concept C_φ as:

$$C_\varphi = P_{\leq p} C_1 \sqcap \dots \sqcap P_{\leq p} C_n \quad \text{with}$$

$$C_i = f(\ell_{i1}) \sqcap \dots \sqcap f(\ell_{ik_i}) \quad \text{where } f(\ell) = \begin{cases} P_{\leq p} X_j & \text{if } \ell = x_j; \\ P_{\leq p}(P_{\leq p} X_j) & \text{if } \ell = \neg x_j. \end{cases}$$

It is standard to verify the following:

Claim. C_φ is satisfiable iff φ is satisfiable.

Proof of the Claim. “ \Rightarrow ”: If C_φ is satisfiable, there is some interpretation $\mathcal{I} = (\Delta^\mathcal{I}, W, (\mathcal{I}_w)_{w \in W}, \mu)$, a domain element $d \in \Delta^\mathcal{I}$ and a world $w \in W$ such that $d \in C_\varphi^{\mathcal{I}, w}$. We define a truth assignment v by taking for $1 \leq j \leq m$

$$v(x_j) = \begin{cases} 1 & \text{if } d \notin (P_{\leq p} X_j)^{\mathcal{I}, w}; \\ 0 & \text{otherwise.} \end{cases}$$

We show that $v(\varphi_i) = 1$ for every clause φ_i in φ and, therefore, $v(\varphi) = 1$. Fix an $i \in \{1, \dots, n\}$. Since $d \in C_\varphi^{\mathcal{I}, w}$, the definition of C_φ yields $p_d^\mathcal{I}(C_i) \leq p$ for all $1 \leq i \leq n$. Since C_i is a conjunction of probabilistic concepts, we must have $p_d^\mathcal{I}(C_i) = 0$. By the semantics, $d \notin f(\ell)^{\mathcal{I}, v}$ for some literal ℓ in φ_i and all worlds $v \in W$, in particular $d \notin f(\ell)^{\mathcal{I}, w}$. We distinguish cases according to the shape of ℓ :

- If $\ell = x_j$, then $d \notin (P_{\leq p} X_j)^{\mathcal{I}, w}$ by the definition of f . By definition of v , we have $v(x_j) = 1$ and, thus, $v(\ell) = 1$ and $v(\varphi_i) = 1$.
- If, on the other hand, $\ell = \neg x_j$, then $d \notin (P_{\leq p}(P_{\leq p} X_j))^{\mathcal{I}, w}$. The semantics yields $d \in (P_{\leq p} X_j)^{\mathcal{I}, w}$. By definition of v , we have $v(x_j) = 0$ and, thus, $v(\ell) = 1$ and $v(\varphi_i) = 1$.

“ \Leftarrow ”: If φ is satisfiable, there is some satisfying truth assignment v , that is, $v(\varphi) = 1$ and, in particular, $v(\varphi_i) = 1$ for all $i \in \{1, \dots, n\}$. Let $\mathcal{I} = (\Delta^\mathcal{I}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ be the probabilistic interpretation defined by taking

- $\Delta^\mathcal{I} = \{d\}$;
- $W = \{w\}$;
- $\mu(w) = 1$;
- $d \in X_j^{\mathcal{I}, w}$ iff $v(x_j) = 1$ for all $1 \leq j \leq m$;

- $C_\varphi^{\mathcal{I},w} = \{d\}$.

We show that \mathcal{I} is a model of C_φ . We start with showing that for every $1 \leq i \leq n$, there is some ℓ in φ_i such that $d \notin f(\ell)^{\mathcal{I},w}$. Since $v(\varphi_i) = 1$ there is some literal ℓ in φ_i such that $v(\ell) = 1$. Again, we distinguish cases according to the form of ℓ :

- If $\ell = x_j$, then $v(x_j) = 1$ and $f(\ell) = P_{\leq p}X_j$. By definition of \mathcal{I} we get $d \in X_j^{\mathcal{I},w}$ and thus $p_d^{\mathcal{I}}(X_j) = 1$. Consequently, we have $d \notin (P_{\leq p}X_j)^{\mathcal{I},w}$.
- If, on the other hand, $\ell = \neg x_j$, then $v(x_j) = 0$ and $f(\ell) = P_{\leq p}(P_{\leq p}X_j)$. By definition of \mathcal{I} we get $d \notin X_j^{\mathcal{I},w}$ and thus $p_d^{\mathcal{I}}(X_j) = 0$. Using the semantics, we obtain $d \notin (P_{\leq p}(P_{\leq p}X_j))^{\mathcal{I},w}$.

Thus, for each $1 \leq i \leq n$, we have $d \notin C_i^{\mathcal{I},w}$. Finally, we obtain $d \in (P_{\leq p}C_i)^{\mathcal{I},w}$ and $d \in C_\varphi^{\mathcal{I},w}$.

It remains to notice that C_φ is unsatisfiable iff $C_\varphi \sqsubseteq B$ for any fresh concept name B . This settles the coNP lower bound. ■

4.3 Complexity of Probabilistic Roles

In this section, we study probabilistic variants of \mathcal{EL} that support *probabilistic roles*. It is known that adding probabilistic roles tends to increase the complexity of reasoning. For instance, in [101], it was shown that the complexity of satisfiability jumps from EXPTIME to (at least) 2EXPTIME-hardness if probabilistic roles are admitted in Prob \mathcal{ALC} . Similarly, subsumption checking in full Prob \mathcal{EL}_{01} was proved PSPACE-hard whereas it is in PTIME when restricting to probabilistic concepts.

4.3.1 Subsumption in full Prob \mathcal{EL}

Let us start with considering full Prob \mathcal{EL} . Notice first that Prob \mathcal{EL} is clearly non-convex as it subsumes the logics considered in the previous section. Along the lines of the reduction from concept satisfiability in \mathcal{ALC} to subsumption in non-convex variants of Prob \mathcal{EL}_c one can actually reduce from satisfiability in Prob \mathcal{ALC} to subsumption in Prob \mathcal{EL} . As Prob \mathcal{ALC} strictly contains Prob \mathcal{EL} , we obtain.

Proposition 4.13. *Subsumption relative to general TBoxes in Prob \mathcal{EL} and concept satisfiability of Prob \mathcal{ALC} concepts relative to general TBoxes are mutually polytime-reducible.*

In particular, since not even decidability is known for Prob \mathcal{ALC} the same holds true for full Prob \mathcal{EL} . On the other hand, we inherit a 2EXPTIME-lower bound from satisfiability in Prob \mathcal{ALC}_{01} . In order to identify well-behaved fragments of Prob \mathcal{EL} , we again restrict

the application of the probabilistic constructors $P_{\sim p}$ and $\exists P_{\sim p}r$. A strategy of restricting the probabilistic constructors that has been successfully applied is the restriction to $P_{>0}$ and $P_{=1}$. One instance is Theorem 3.4 from the previous chapter. Other examples are the restriction $\text{Prob}\mathcal{ALC}_{01}$ of $\text{Prob}\mathcal{ALC}$ [101] and a similar restriction of probabilistic CTL [24]. In the former, satisfiability in $\text{Prob}\mathcal{ALC}_{01}$ was shown to be 2EXPTIME-complete; in the latter, satisfiability in probabilistic CTL restricted to $P_{>0}$ and $P_{=1}$ was shown to be EXPTIME-complete. However, in both cases, decidability of the “full” logic remains open. Thus, it is interesting to study the fragment $\text{Prob}\mathcal{EL}_{01}$.

Let us first point out that $\text{Prob}\mathcal{EL}_{01}$ is in a sense the maximal fragment that in principle might have a complexity lower than 2EXPTIME. Indeed, consider the logic $\text{Prob}\mathcal{EL}_{01}^{\sim p}$ featuring another probabilistic constructor $P_{\sim p}$ (on concepts or roles). With the techniques presented in the previous section, we can easily show that this logic is non-convex. Thus, we can reduce from concept satisfiability relative to general $\text{Prob}\mathcal{ALC}_{01}$ -TBoxes to checking subsumption relative to $\text{Prob}\mathcal{EL}_{01}^{\sim p}$ -TBoxes and obtain:

Theorem 4.14. *For all $\sim \in \{<, \leq, =, \geq, >\}$ and $p \in (0, 1)$, checking subsumption relative to general $\text{Prob}\mathcal{EL}_{01}^{\sim p}$ -TBoxes is 2EXPTIME-hard.*

4.3.2 Subsumption in $\text{Prob}\mathcal{EL}_{01}$ is PSPACE-complete

As already mentioned, there were reasons to believe that subsumption in $\text{Prob}\mathcal{EL}_{01}$ is actually 2EXPTIME-complete: so far any two-dimensional extension of \mathcal{EL} turned out to have the same complexity as the corresponding extension of the expressive DL \mathcal{ALC} , see for example [5, 52]. However, we show here that this is not the case by proving a PSPACE upper bound, thus establishing PSPACE-completeness for both positive and unrestricted subsumption.

It is instructive to have a look at the following example which realizes an exponential counter in $\text{Prob}\mathcal{EL}_{01}$ and thus demonstrates the expressive power of $\text{Prob}\mathcal{EL}_{01}$ and the way inferences are made. Notice that the example is inspired by the PSPACE lower bound for $\text{Prob}\mathcal{EL}_{01}$ from [101].

Example 4.15. *Fix a positive integer n . We encode a number $0 \leq k < 2^n$ as follows:*

$$(X = k) = \bigcap_{i=0}^{n-1} \begin{cases} X_i & i\text{-th bit of } k \text{ is } 1; \\ \overline{X_i} & \text{otherwise,} \end{cases}$$

where $X_i, \overline{X_i}$ represent the bits of a binary counter. The goal of this example is to give a (polynomially sized) TBox $\mathcal{T}_n, n > 0$ such that $\mathcal{T}_n \models A \sqsubseteq P_{>0}(X = k)$ for any $0 \leq k < 2^n$.

Let $\alpha = P_{>0}r$ and let the TBox \mathcal{T}_n consist of the following set of axioms (the colors

will be used in an illustration below):

$$\begin{array}{ll}
 A \sqsubseteq \exists\alpha.A & A \sqsubseteq P_{>0}(X=0) \\
 \exists\alpha.X_0 \sqsubseteq \overline{X_0} \sqcap C_1 & \exists\alpha.\overline{X_0} \sqsubseteq X_0 \sqcap \overline{C_1} \\
 C_i \sqcap \exists\alpha.X_i \sqsubseteq \overline{X_i} \sqcap C_{i+1} & C_i \sqcap \exists\alpha.\overline{X_i} \sqsubseteq X_i \sqcap \overline{C_{i+1}} \text{ for all } 0 < i < n \\
 \overline{C_i} \sqcap \exists\alpha.X_i \sqsubseteq X_i \sqcap \overline{C_{i+1}} & \overline{C_i} \sqcap \exists\alpha.\overline{X_i} \sqsubseteq \overline{X_i} \sqcap \overline{C_{i+1}} \text{ for all } 0 < i < n.
 \end{array}$$

Intuitively, the first axiom enforces an infinite chain of A -elements. The second axiom enforces that for each element of the chain, there is some world where this element satisfies $(X=0)$. The last 6 concept inclusions imply the inclusion $\exists\alpha.(X=i) \sqsubseteq (X=i+1)$ (using auxiliary concept names $C_i, \overline{C_i}$ representing the carry bits).

Now, fix some $0 \leq k < 2^n$. By the first axiom, we have that $\mathcal{T}_n \models A \sqsubseteq \exists\alpha^k.A$. By the second axiom, we get $\mathcal{T}_n \models A \sqsubseteq \exists\alpha^k.P_{>0}(X=0)$. By the semantics, we have $\mathcal{T}_n \models A \sqsubseteq P_{>0}(\exists\alpha^k.(X=0))$. Using the remaining axioms it is not hard to see that $\mathcal{T}_n \models A \sqsubseteq P_{>0}(X=k)$; see also Figure 4.3 for an illustration.

On a high level, reasoning in Example 4.15 is performed in three steps:

- (i) select (what we will later call) a ‘trace’, i.e., a sequence of probabilistic roles that is implied by some concept name;
- (ii) select some concept of the form $P_{>0}B$ implied by the ‘end’ of the trace (inducing a fresh world);
- (iii) use the TBox to propagate information back to the ‘start’ of the trace.

In Example 4.15, we select in the first step α^k for some k for the concept name A . In the second step, we note that A (the ‘end’ of the trace) implies $P_{>0}(X=0)$. In step three, we repeatedly apply the last six concept inclusions to obtain the result. In order to devise an algorithm for deciding subsumption in $\text{Prob}\mathcal{EL}_{01}$, we certainly have to take care of the kind of reasoning described above. Even better, it turns out that the pattern we identified in a way already captures all the inferences. In particular, we will rigorously define *traces* and use them as the main tool.

In what follows, we specify a non-deterministic consequence-driven algorithm that decides subsumption and can be implemented in polynomial space. Without loss of generality, we assume that the TBox is in the following normal form. A *basic concept* is a concept of the form \top , A , $P_{>0}A$, $P_{=1}A$, or $\exists\alpha.A$, where A is a concept name and α is a *role*, i.e., of the form r , $P_{>0}r$, or $P_{=1}r$ with r a role name. In the latter two cases we call the role α a *probabilistic role*. Now, every concept inclusion in the input TBox is required to be of the form

$$X_1 \sqcap \dots \sqcap X_n \sqsubseteq X,$$

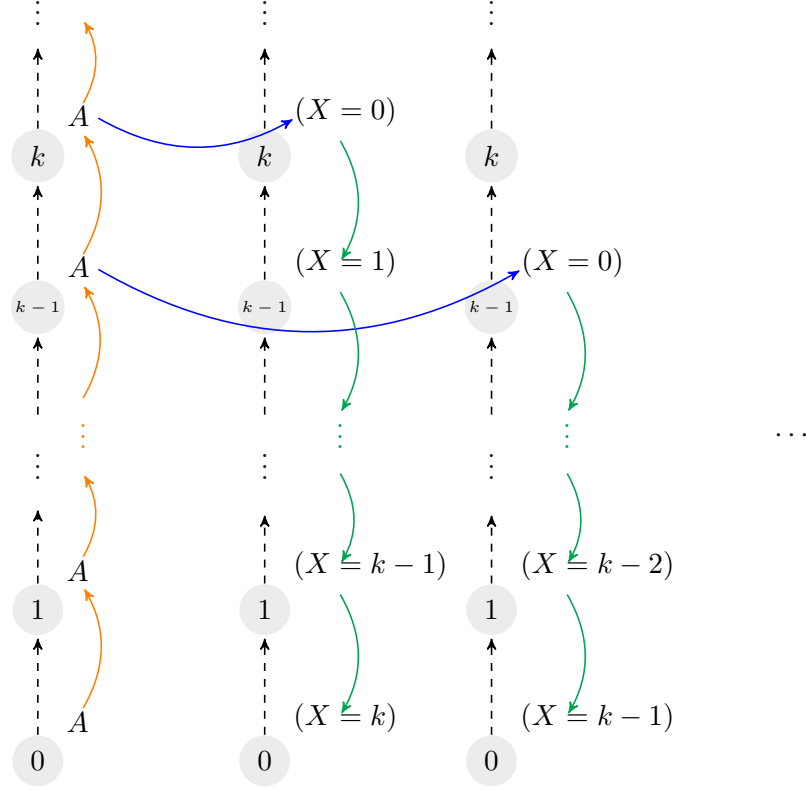


Figure 4.3: Illustration of the reasoning in Example 4.15. Each thread of elements $0, 1, \dots$ represents one possible world; dashed arrows indicate a probabilistic role $P_{>0}r$ between two elements. Colored arrows indicate the application of concept inclusions: orange corresponds to the first CI, blue to the second CI, and green to the group of the 6 remaining CIs.

with X_1, \dots, X_n, X basic concepts. It is not hard to show that every TBox can be transformed into this normal form in polynomial time such that (non-)subsumption between the concept names that occur in the original TBox is preserved. We refrain from giving details here, but refer the reader to similar constructions, e.g., in [10].

Let \mathcal{T} be the input TBox in normal form, \mathbf{CN} the set of concept names that occur in \mathcal{T} , \mathbf{BC} the set of basic concepts in \mathcal{T} , and \mathbf{ROL} the set of roles in \mathcal{T} . Our algorithm maintains the following data structures:

- a mapping Q that associates with each $A \in \mathbf{CN}$ a subset $Q(A) \subseteq \mathbf{BC}$ such that $\mathcal{T} \models A \sqsubseteq X$ for all $X \in Q(A)$;
- a mapping Q_{cert} that associates with each $A \in \mathbf{CN}$ a subset $Q_{\text{cert}}(A) \subseteq \mathbf{BC}$ such

R1	If $X_1 \sqcap \dots \sqcap X_n \sqsubseteq X \in \mathcal{T}$ and $X_1, \dots, X_n \in \Gamma$, then add X to Γ .
R2	If $P_{=1}A \in \Gamma$, then add A to Γ .
R3	If $A \in \Gamma$, then add $P_{>0}A$ to Γ .
R4	If $\exists P_{=1}r.A \in \Gamma$, then add $\exists r.A$ and $\exists P_{>0}r.A$ to Γ .
R5	If $\exists r.A \in \Gamma$, then add $\exists P_{>0}r.A$ to Γ .
R6	If $\exists \alpha.A \in \Gamma$ and $B \in Q(A)$, then add $\exists \alpha.B$ to Γ .

Figure 4.4: Saturation rules for $\text{cl}(\Gamma)$.

that $\mathcal{T} \models A \sqsubseteq P_{=1}X$ for all $X \in Q_{\text{cert}}(A)$;

- a mapping R that associates with each probabilistic role $\alpha \in \text{ROL}$ a binary relation $R(\alpha)$ on CN such that $\mathcal{T} \models A \sqsubseteq P_{>0}(\exists \alpha.B)$ for all $(A, B) \in R(\alpha)$.

Some intuition about the data structures is already provided above; e.g., $X \in Q(A)$ means that $\mathcal{T} \models A \sqsubseteq X$. However, there is also another view on these structures that will be important in what follows: they represent an abstract view of a model of \mathcal{T} , where each set $Q(A)$ describes the concept memberships of a domain element d in a world w with $d \in A^{\mathcal{T},w}$ and R describes the structure of the rigid roles, i.e., when $(A, B) \in R(\alpha)$, then $d \in A^{\mathcal{T},w}$ implies that in some world v with positive probability, d has an element described by $Q(B)$ as an α -successor. In this context, $Q_{\text{cert}}(A)$ contains all concepts that are *certain* for any domain element that satisfies A in *some* world, i.e., all concepts that hold with probability 1. Note that non-probabilistic roles r are not represented in the $R(\cdot)$ data structure; they are treated in the basic concepts. The data structures are initialized for all $A \in \text{CN}$ and probabilistic roles α :

$$Q(A) = \{\top, A\}; \quad Q_{\text{cert}}(A) = \{\top\}; \quad R(\alpha) = \emptyset.$$

The sets $Q(\cdot)$, $Q_{\text{cert}}(\cdot)$, and $R(\cdot)$ are then repeatedly extended by the application of various rules. Before we can introduce these rules, we need some preliminaries. As the first step, Figure 4.4 presents a (different!) set of rules that serves the purpose of saturating a set of concepts Γ . We use $\text{cl}(\Gamma)$ to denote the set of concepts that is the result of exhaustively applying the displayed rules to Γ , where any rule can only be applied if the added concept is in BC , but not yet in Γ . The rules access the data structure $Q(\cdot)$ introduced above and shall later be applied to the sets $Q(A)$ and $Q_{\text{cert}}(A)$, but they will also serve other purposes as described below. It is not hard to see that rule application terminates after polynomially many steps.

The rules that are used for completing the data structures $Q(\cdot)$, $Q_{\text{cert}}(\cdot)$, and $R(\cdot)$ are more complex and refer to ‘traces’ through these data structures, which were already motivated and which we formally define next.

Definition 4.16 (Trace). A trace to A_n is a finite sequence $S, A_1, \alpha_2, A_2, \dots, \alpha_n, A_n$ where

1. each $A_i \in \mathbf{CN}$ and each $\alpha_i \in \mathbf{ROL}$ is a probabilistic role;
2. $S = B$ for some $P_{>0}B \in Q(A_1)$ or $S = (r, B)$ for some $(A_1, B) \in R(P_{>0}r)$;
3. $(A_i, A_{i-1}) \in R(\alpha_i)$ for $1 < i \leq n$.

If t is a trace $S, A_1, \alpha_2, A_2, \dots, \alpha_n, A_n$ and $k \leq n$, we use t_k to denote the trace $S, A_1, \alpha_2, \dots, \alpha_k, A_k$. Intuitively, the purpose of a trace is to deal with worlds that are generated by concepts $P_{>0}A$ and $\exists P_{>0}r.A$; there can be infinitely many such worlds as $\text{Prob}\mathcal{EL}_{01}$ lacks the finite model property, see [101]. The trace starts at some domain element represented by a set $Q(A_1)$ in the world generated by the first element S of the trace, then repeatedly follows role edges represented by $R(\cdot)$ backwards until it reaches the final domain element represented by $Q(A_n)$. The importance of traces stems from the fact that information can be propagated along them, as illustrated by Example 4.15 and formally captured by the following notion. Note that the rules **R1** to **R6** are used in every step of this inductive definition.

Definition 4.17 (Type of a trace). Let $t = S, A_1, \alpha_2, \dots, \alpha_n, A_n$ be a trace. Then the type $\Gamma(t) \subseteq \mathbf{BC}$ of t is defined as:

$$\Gamma(t) = \begin{cases} \text{cl}(\{B\} \cup Q_{\text{cert}}(A_1)) & \text{if } t = (B, A_1); \\ \text{cl}(Q_{\text{cert}}(A_1) \cup \{\exists r.B' \in \mathbf{BC} \mid B' \in Q_{\text{cert}}(B)\}) & \text{if } t = ((r, B), A_1); \\ \text{cl}(Q_{\text{cert}}(A_n) \cup \{\exists \alpha_n.B' \in \mathbf{BC} \mid B' \in \Gamma(t_{n-1})\}) & \text{if } n > 1. \end{cases}$$

Figure 4.5 shows the rules used for completing the data structures $Q(\cdot)$, $Q_{\text{cert}}(\cdot)$, and $R(\cdot)$. Rules **S1** to **S5** are rather straightforward and do not require further explanation. Particularly interesting are rules **S6** and **S7** because they implement the propagation of information along traces, as announced above: if there is a trace t to B , then any domain element that satisfies B in *some* world must satisfy the concepts in $\Gamma(t)$ in some other world. So if for example $P_{>0}A \in \Gamma(t)$, we need to add $P_{>0}A$ also to $Q_{\text{cert}}(B)$. Precisely this is captured in rule **S6**. In a similar way one can explain **S7**.

Our algorithm for deciding subsumption starts with the initial data structures defined above and then exhaustively applies the rules shown in Figure 4.5. To decide whether $\mathcal{T} \models A \sqsubseteq B$, it then simply checks whether $B \in Q(A)$ after the algorithm terminated.

Lemma 4.18. Let \mathcal{T} be a general $\text{Prob}\mathcal{EL}_{01}$ -TBox in normal form and A_0 and B_0 be concept name and a basic concept, respectively. Then $\mathcal{T} \models A_0 \sqsubseteq B_0$ iff, after exhaustive rule application, $B_0 \in Q(A_0)$.

- S1** Apply **R1-R6** to $Q(A)$ and $Q_{\text{cert}}(A)$.

S2 If $P_*B \in Q(A)$,
then add P_*B to $Q_{\text{cert}}(A)$.

S3 If $C \in Q_{\text{cert}}(A)$,
then add $P_{=1}C$ and C to $Q(A)$.

S4 If $\exists\alpha.B \in Q(A)$ with α a probabilistic role,
then add (A, B) to $R(\alpha)$.

S5 If $(A_1, A_2) \in R(\alpha)$, $B \in Q_{\text{cert}}(A_2)$,
then add $\exists\alpha.B$ to $Q_{\text{cert}}(A_1)$.

S6 If t is a trace to B and $P_*A \in \Gamma(t)$,
then add P_*A to $Q_{\text{cert}}(B)$.

S7 If t is a trace to B and $\exists\alpha.A \in \Gamma(t)$ with α a probabilistic role,
then add (B, A) to $R(\alpha)$.

Figure 4.5: The rules for completing the data structures.

Proof. For the “if” direction we show that the following invariants of the algorithm hold:

$$C \in Q(A) \text{ implies } \mathcal{T} \models A \sqsubseteq C \quad (\text{inv1})$$

$$C \in Q_{\text{cert}}(A) \text{ implies } \mathcal{T} \models A \sqsubseteq P_{=1}C \quad (\text{inv2})$$

$$(A, B) \in R(\alpha) \text{ implies } \mathcal{T} \models A \sqsubseteq P_{>0}(\exists\alpha.B) \quad (\text{inv3})$$

The proof is by induction on the number of applications of the rules in Figure 4.5. The induction base is trivial since $A \sqsubseteq A$ and $A \sqsubseteq \top$. For showing the induction step, we use sets of concepts Γ to denote the conjunction $\bigwedge_{C \in \Gamma} C$. We start by showing soundness of the rules **R1-R6**. In particular, for every set of concepts Γ it holds

$$\mathcal{T} \models \Gamma \sqsubseteq \text{cl}(\Gamma). \quad (*)$$

This fact is a direct consequence of the semantics for rules **R1-R5**. For **R6** assume $\exists\alpha.A \in \Gamma$ and $B \in Q(A)$. Invariant **(inv1)** implies $\mathcal{T} \models A \sqsubseteq B$, which means that we can certainly add $\exists\alpha.B$ to Γ .

Next, we analyze traces closer and prove the following claim.

Claim 1. If t is a trace to B , then $\mathcal{T} \models B \sqsubseteq P_{>0}(\Gamma(t))$.

Proof of Claim 1. Let $t = S, A_1, \alpha_2, \dots, \alpha_n, A_n$. The proof is by induction on the length n of t . For the induction base, we let $n = 1$ and distinguish cases according to the form

of S . Consider first the case that the trace starts with $S = B$, i.e., $P_{>0}B \in Q(A_1)$. From invariants **(inv1)** and **(inv2)**, it follows that $\mathcal{T} \models A_1 \sqsubseteq P_{>0}(B \sqcap Q_{\text{cert}}(A_1))$. Since $\Gamma(t) = \text{cl}(\{B\} \cup Q_{\text{cert}}(A_1))$ we obtain $\mathcal{T} \models A_1 \sqsubseteq P_{>0}(\Gamma(t))$, by Equation (*).

Assume now that the trace starts with $S = (r, B)$, i.e., $(A_1, B) \in R(P_{>0}r)$. By **(inv3)**, we have $\mathcal{T} \models A_1 \sqsubseteq P_{>0}(\exists P_{>0}r.B)$, thus $\mathcal{T} \models A_1 \sqsubseteq P_{>0}(\exists r.P_{>0}B)$. From invariant **(inv2)**, we get $\mathcal{T} \models A_1 \sqsubseteq P_{=1}(Q_{\text{cert}}(A_1))$ and $\mathcal{T} \models B \sqsubseteq P_{=1}(Q_{\text{cert}}(B))$. Overall, we obtain:

$$\mathcal{T} \models A_1 \sqsubseteq P_{>0}(Q_{\text{cert}}(A_1) \sqcap \exists r.Q_{\text{cert}}(B)).$$

Since $\Gamma(t) = \text{cl}(Q_{\text{cert}}(A_1) \cup \{\exists r.B' \mid B' \in Q_{\text{cert}}(B)\})$, we can apply (*) to obtain:

$$\mathcal{T} \models A_1 \sqsubseteq P_{>0}(\Gamma(t)).$$

For the induction step, let $n > 1$. By Definition 4.16, $(A_n, A_{n-1}) \in R(\alpha_n)$, thus, invariant **(inv3)** yields $\mathcal{T} \models A_n \sqsubseteq P_{>0}(\exists \alpha_n.A_{n-1})$. Applying the induction hypothesis, we get

$$\mathcal{T} \models A_n \sqsubseteq P_{>0}(\exists \alpha_n.P_{>0}(\Gamma(t_{n-1}))).$$

Since $\exists \alpha_n.P_{>0}C \sqsubseteq P_{>0}\exists \alpha_n.C$ is valid for all C and probabilistic roles α_n , we obtain

$$\mathcal{T} \models A_n \sqsubseteq P_{>0}(\exists \alpha_n.\Gamma(t_{n-1})).$$

On the other hand, **(inv2)** implies $\mathcal{T} \models A_n \sqsubseteq P_{=1}(Q_{\text{cert}}(A_n))$. Together this yields:

$$\mathcal{T} \models A_n \sqsubseteq P_{>0}(Q_{\text{cert}}(A_n) \sqcap \exists \alpha_n.\Gamma(t_{n-1})).$$

Since $\Gamma(t) = \text{cl}(Q_{\text{cert}}(A_n) \cup \{\exists \alpha_n.B \mid B \in \Gamma(t_{n-1})\})$, we can apply (*) to finally get

$$\mathcal{T} \models A_n \sqsubseteq P_{>0}(\Gamma(t)).$$

This finishes the proof of the claim.

It remains to show that the rules in Figure 4.5 preserve the invariants:

- S1** Direct consequence of (*).
- S2** Since $P_{>0}B \sqsubseteq P_{=1}(P_{>0}B)$ and $P_{=1}B \sqsubseteq P_{=1}(P_{=1}B)$ are valid concept inclusions this is a direct consequence of the semantics.
- S3** $C \in Q_{\text{cert}}(A)$ implies $\mathcal{T} \models A \sqsubseteq P_{=1}C$ by invariant **(inv2)**, hence also $\mathcal{T} \models A \sqsubseteq C$.
- S4** $\exists \alpha.B \in Q(A)$ implies $\mathcal{T} \models A \sqsubseteq \exists \alpha.B$ by invariant **(inv1)**, thus also $\mathcal{T} \models A \sqsubseteq P_{>0}(\exists \alpha.B)$.
- S5** On the one hand, $(A_1, A_2) \in R(\alpha)$ implies $\mathcal{T} \models A_1 \sqsubseteq P_{>0}(\exists \alpha.A_2)$, by **(inv3)**. On the other hand, $B \in Q_{\text{cert}}(A_2)$ yields $\mathcal{T} \models A_2 \sqsubseteq P_{=1}B$, by **(inv2)**. By the semantics, together they imply $\mathcal{T} \models A_1 \sqsubseteq P_{=1}(\exists \alpha.B)$.

S6 Let t be a trace to B and $\Gamma = \Gamma(t)$ its type. By the above claim, $\mathcal{T} \models B \sqsubseteq P_{>0}C$ for every $C \in \Gamma$. Thus, in particular, $\mathcal{T} \models B \sqsubseteq P_*A$ for all $P_*A \in \Gamma$. By the semantics, $\mathcal{T} \models B \sqsubseteq P_{=1}(P_*A)$, so P_*A can be added to $Q_{\text{cert}}(B)$.

S7 Analogously to **S6**.

Assume now $B_0 \in Q(A_0)$. Invariant (**inv1**) implies $\mathcal{T} \models A_0 \sqsubseteq B_0$ which finishes the proof of the “if”-direction.

For the “only if”-direction, we provide a probabilistic model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W})$ of \mathcal{T} such that there is a world $w \in W$ and a domain element $d \in \Delta^{\mathcal{I}}$ with $d \in A_0^{\mathcal{I},w}$ but $d \notin B_0^{\mathcal{I},w}$.

We define sequences $\Delta^{\mathcal{I}_0}, \Delta^{\mathcal{I}_1}, \dots, W_0, W_1, \dots$, and partial maps π_0, π_1, \dots with $\pi_i : \Delta^{\mathcal{I}_i} \times W_i \rightarrow 2^{\text{BC}}$. Our desired sets $\Delta^{\mathcal{I}}$ and W are then obtained in the limit. The elements of the sets $\Delta^{\mathcal{I}_i}$ are sequences of triples (α, w, A) where $\alpha \in \text{ROL}$ is a role, $w \in W_i$ is a world, and $A \in \text{CN}$ is a concept name. For such a sequence σ , we use σ_j to denote the prefix of σ that consists of the first j triples.

Intuitively, the worlds of \mathcal{I} correspond to traces. In particular, all worlds (except two initial ones) will be of the form (σ, S) for some $\sigma \in \Delta^{\mathcal{I}}$ and S either B or (r, B) for some concept name B and role name r . For establishing the close correspondence between worlds and traces, we define a function δ that maps worlds (σ, S) to the sequence $S, A_n, \hat{\alpha}_n, \dots, \hat{\alpha}_2, A_1$ where $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n)$ and $\hat{\alpha} = \alpha$ if α is a probabilistic role and $\hat{\alpha} = P_{>0}r$ if α is the role name r . Note that δ reverses the order of the A_i and the α_i . We will show that the world w precisely witnesses the existence of the trace $\delta(w)$.

To start the construction of \mathcal{I} , set

- $\Delta^{\mathcal{I}_0} = \{(\alpha, \varepsilon, A_0)\}$ where α is any role (not important) and A_0 is the concept name from the left-hand side of the subsumption which is to be checked;
- $W_0 = \{\varepsilon, 0\}$,
- $\pi((\alpha, \varepsilon, A_0), \varepsilon) = Q(A_0)$ and $\pi((\alpha, \varepsilon, A_0), 0) = Q_{\text{cert}}(A_0)$.

For the induction step, we start with setting $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_{i-1}}$, $W_i = W_{i-1}$, and $\pi_i = \pi_{i-1}$, and then apply the following rules:

1. If $\exists \alpha.A \in \pi_i(\sigma, w)$ for some $\sigma \in \Delta^{\mathcal{I}_i}$ and $w \in W_i$, then set $\sigma' := \sigma \cdot (\alpha, w, A)$ and
 - (a) add σ' to $\Delta^{\mathcal{I}_i}$ (if it does not exist yet);
 - (b) set $\pi_i(\sigma', w) = Q(A)$ and $\pi_i(\sigma', v) = Q_{\text{cert}}(A)$ for all $v \in W_i \setminus \{w\}$.
2. If $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta^{\mathcal{I}_i}$ and $P_{>0}B \in Q(A_n)$, then
 - (a) add (σ, B) to W_i (if it does not exist yet);

- (b) set $\pi_i(\sigma_j, (\sigma, B)) = \Gamma(\delta(\sigma, B)_{n-j+1})$ for all $1 \leq j \leq n$; and
 - (c) set $\pi_i(\sigma' \cdot (\alpha, w, A), (\sigma, B)) = Q_{\text{cert}}(A)$ for all $\sigma' \cdot (\alpha, w, A) \in \Delta^{\mathcal{I}_i}$ that are *not* a prefix of σ .
3. If $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta^{\mathcal{I}_i}$ and $(A_n, B) \in R(P_{>0}r)$, then
- (a) add (σ, r, B) to W_i (if it does not exist yet);
 - (b) set $\pi_i(\sigma_j, (\sigma, r, B)) = \Gamma(\delta(\sigma, (r, B))_{n-j+1})$ for all $1 \leq j \leq n$; and
 - (c) set $\pi_i(\sigma' \cdot (\alpha, w, A), (\sigma, r, B)) = Q_{\text{cert}}(A)$ for all $\sigma' \cdot (\alpha, w, A) \in \Delta^{\mathcal{I}_i}$ that are *not* a prefix of σ .

Finally, set $\Delta^{\mathcal{I}} = \bigcup_{i \geq 0} \Delta^{\mathcal{I}_i}$, $W = \bigcup_{i \geq 0} W_i$, and $\pi = \bigcup_{i \geq 0} \pi_i$. Define μ such that $\mu(w) > 0$ for all $w \in W$ and $\sum_{w \in W} \mu(w) = 1$. If W is finite this is clearly possible; otherwise assign the probabilities $1/2, 1/4, 1/8, \dots$ to (an enumeration of) the worlds. It remains to define the interpretation of concept and role names:

$$\begin{aligned}
 A^{\mathcal{I}, w} &= \{\sigma \in \Delta^{\mathcal{I}} \mid A \in \pi(\sigma, w)\}; \\
 r^{\mathcal{I}, w} &= \{(\sigma, \sigma \cdot (P_{>0}r, v, A)) \mid \sigma \cdot (P_{>0}r, v, A) \in \Delta^{\mathcal{I}}, w = (\sigma, r, A)\} \cup \\
 &\quad \{(\sigma, \sigma \cdot (r, w, A)) \mid \sigma \cdot (r, w, A) \in \Delta^{\mathcal{I}}\} \cup \\
 &\quad \{(\sigma, \sigma \cdot (P_{=1}r, v, A)) \mid \sigma \cdot (P_{=1}r, v, A) \in \Delta^{\mathcal{I}}\}.
 \end{aligned}$$

In the following claim we state and prove some important properties of the construction. Note that Points (i) and (ii) verify that the construction, specifically steps 2(b) and 3(b) are well-defined.

Claim 2. The following points hold:

- (i) for every $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta^{\mathcal{I}}$ we have $(A_j, A_{j+1}) \in R(\hat{\alpha}_{j+1})$ for all $1 \leq j < n$;
- (ii) for every $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta^{\mathcal{I}}$ and $w \in W$ we have $\pi(\sigma, w)$ is either $Q(A_n)$, $Q_{\text{cert}}(A_n)$ or $\Gamma(t)$ for some trace t to A_n ;
- (iii) for every $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta^{\mathcal{I}}$, there are worlds w, v with $\pi(\sigma, w) = Q(A_n)$ and $\pi(\sigma, v) = Q_{\text{cert}}(A_n)$;
- (iv) $P_*A \in \pi(\sigma, w)$ if and only if $P_*A \in \pi(\sigma, v)$ for all $\sigma \in \Delta^{\mathcal{I}}$ and $w, v \in W$;
- (v) for all probabilistic roles α and $\sigma, \sigma' \in \Delta^{\mathcal{I}}$ with $\sigma' = \sigma \cdot (\alpha, v, B)$ and $A \in \pi(\sigma', w)$ we have $\exists \alpha.A \in \pi(\sigma, w)$.

Proof of Claim 2. Throughout the proof we assume that $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n)$. We prove Points (i) and (ii) together by induction on the number of rule applications. The induction base is immediate by the definition of $\Delta^{\mathcal{I}_0}$, W_0 , and π_0 . For the induction step assume first rule 1 is applied to σ , i.e., $\exists \alpha_{n+1}. A_{n+1} \in \pi_i(\sigma, w)$ for some $w \in W_i$. By induction hypothesis of Point (ii), $\pi_i(\sigma, w)$ is either $Q(A_n)$, $Q_{\text{cert}}(A_n)$, or $\Gamma(t)$ for some trace t to A_n . Thus, $\pi_i(\sigma, w)$ is closed under **cl** and **R5** yields $\exists \hat{\alpha}_{n+1}. A_{n+1} \in \pi_i(\sigma, w)$. Now, rules **S4**, **S3**, and **S7** imply $(A_n, A_{n+1}) \in R(\hat{\alpha}_n)$, i.e., Point (i) is preserved after application of rule 1. Assume now that rule 2 is applied to σ and let $P_{>0}B \in Q(A_n)$. By this fact and induction hypothesis for Point (i), it is immediately clear that $t := \delta(\sigma, B)$ is a trace. Moreover, one can easily check that t_{n-j+1} is a trace to A_j for all $1 \leq j \leq n$. Hence, $\pi(\sigma_j, (\sigma, B))$ is the type of a trace to A_j , namely $\Gamma(t_{n-j+1})$. All other $\pi(\cdot, (\sigma, B))$ are set to $Q_{\text{cert}}(A)$ for the correct A . Thus, rule 2 preserves Point (ii). Similarly, it can be shown that also rule 3 preserves Point (ii).

Point (iii) can be proved by induction on the number of rule applications. The induction base is clear by the definition of \mathcal{I}_0 . For the induction step it suffices to look at rule 1 and observe that when σ is added to $\Delta^{\mathcal{I}}$, we set $\pi(\sigma, w) = Q(A_n)$ for one world w and $\pi(\sigma, v) = Q_{\text{cert}}(A_n)$ for all other worlds v (and there exist at least two worlds).

For Point (iv) we make a case distinction on $\pi(\sigma, w)$ and $\pi(\sigma, v)$. By Point (ii), both are either $Q(A_n)$, $Q_{\text{cert}}(A_n)$ or $\Gamma(t)$ for some trace t to A_n . For symmetry reasons it suffices to consider $\pi(\sigma, w)$. If $\pi(\sigma, w) = Q_{\text{cert}}(A_n)$, then $P_*A \in Q(A)$ (by rule **S3**) and $P_*A \in \Gamma(t)$ for any trace t to A_n , by Definition 4.17. If $\pi(\sigma, w) = Q(A_n)$, then by rule **S2**, P_*A will be in $Q_{\text{cert}}(A_n)$ and we proceed as before. If $\pi(\sigma, w) = \Gamma(t)$ for some trace t to A_n , then by **S6** we have $P_*A \in Q_{\text{cert}}(A_n)$ and, again, we can continue as before.

For proving Point (v), we make a case distinction on $\pi(\sigma', w)$ according to Point (ii).

- If $\pi(\sigma', w) = Q(B)$, then construction rule 1 implies $\exists \alpha. B \in \pi(\sigma, w)$. Since $\pi(\sigma, w)$ is closed under **cl**, by **R6** we obtain $\exists \alpha. A \in \pi(\sigma, w)$.
- Assume $\pi(\sigma', w) = Q_{\text{cert}}(B)$. By Point (i), $(A_n, B) \in R(\alpha)$ (note that α is probabilistic). Now, **S5** implies $\exists \alpha. A \in Q_{\text{cert}}(A_n) \subseteq \pi(\sigma, w)$.
- Assume $\pi(\sigma', w) = \Gamma(t)$ for some trace t to B . Thus, $\pi(\sigma, w)$ is also defined as the type $\Gamma(t')$ of some trace t' . More precisely, $t' = t, \alpha, B'$ for some B' . By definition of the type of trace t' , we get $\exists \alpha. A \in \pi(\sigma, w)$.

This finishes the proof of Claim 2 and we are ready to show the ‘truth lemma’ of our model construction.

Claim 3. For all $\sigma \in \Delta^{\mathcal{I}}$, $w \in W$, and $C \in \text{BC}$, we have $\sigma \in C^{\mathcal{I}, w}$ iff $C \in \pi(\sigma, w)$.

Proof of Claim 3. We prove the claim by a case distinction on the form of C . Throughout the following we assume $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n)$.

- $C = \top$. Then both $\sigma \in \top^{\mathcal{I},w}$ and $\top \in \pi(\sigma, w)$ for all $\sigma \in \Delta^{\mathcal{I}}$ and $w \in W$.
- $C = A \in \text{CN}$. For this case, the claim holds trivially by definition of the interpretation of concept names.
- $C = P_{>0}A$. “if”: Let $\sigma \in (P_{>0}A)^{\mathcal{I},w}$. Then, by the semantics, $\sigma \in A^{\mathcal{I},v}$ for some $v \in W$. Induction hypothesis implies $A \in \pi(\sigma, v)$. By **R3**, also $P_{>0}A \in \pi(\sigma, v)$, and by Claim 2(iv), $P_{>0}A \in \pi(\sigma, w)$.
“only if”: Let $P_{>0}A \in \pi(\sigma, w)$. By Claim 2(iii), there is some world v with $\pi(\sigma, v) = Q(A_n)$. By Claim 2(iv), $P_{>0}A \in Q(A_n)$. By construction rule 2(a), the world $v' = (\sigma, A)$ is in W . By step (b) of rule 2, $A \in \pi(\sigma, v') = \Gamma(\delta(v')_1)$. Induction hypothesis yields $\sigma \in A^{\mathcal{I},v'}$, thus $\sigma \in (P_{>0}A)^{\mathcal{I},w}$.
- $C = P_{=1}A$. “if”: Let $\sigma \in (P_{=1}A)^{\mathcal{I},w}$, thus $\sigma \in A^{\mathcal{I},v}$ for all $v \in W$. By induction hypothesis, $A \in \pi(\sigma, v)$ for all $v \in W$. By Claim 2(iii), there is a world v' such that $\pi(\sigma, v') = Q_{\text{cert}}(A_n)$; thus, $A \in Q_{\text{cert}}(A_n)$. By **S3**, $P_{=1}A \in Q(A_n)$, and by **S2** also $P_{=1}A \in Q_{\text{cert}}(A_n)$. By Claim 2(iv) we obtain $P_{=1}A \in \pi(\sigma, w)$.
“only if”: Let $P_{=1}A \in \pi(\sigma, w)$. By Claim 2(iv), $P_{=1}A \in \pi(\sigma, v)$ for all $v \in W$. Since all $\pi(\sigma, v)$ are closed under cl, **R2** implies $A \in \pi(\sigma, v)$ for all v . By the hypothesis, $\sigma \in A^{\mathcal{I},v}$ for all $v \in W$, which, by the semantics, implies $\sigma \in (P_{=1}A)^{\mathcal{I},w}$.
- $C = \exists r.A$. “if”: Assume $\sigma \in (\exists r.A)^{\mathcal{I},w}$. By the semantics, there is a $\sigma' \in \Delta^{\mathcal{I}}$ such that $\sigma' \in A^{\mathcal{I},w}$ and $(\sigma, \sigma') \in r^{\mathcal{I},w}$. By induction hypothesis, we know that $A \in \pi(\sigma', w)$. Due to the model construction, there are three possibilities for (σ, σ') being in $r^{\mathcal{I},w}$:
 - $\sigma' = \sigma \cdot (P_{>0}r, v, B)$ and $w = (\sigma, r, B)$ for some concept name B . By construction rule 3(c), $\pi(\sigma', w) = Q_{\text{cert}}(B)$ since σ' is not a prefix of σ . Hence, $A \in Q_{\text{cert}}(B)$. By rule 3(b), we have that $\pi(\sigma, w) = \Gamma(\delta(w)_1) = \text{cl}(Q_{\text{cert}}(A_n) \cup \{\exists r.B' \mid B' \in Q_{\text{cert}}(B)\})$. Since $A \in Q_{\text{cert}}(B)$, $\exists r.A \in \pi(\sigma, w)$.
 - $\sigma' = \sigma \cdot (r, w, B)$ for some B . By construction, in particular rule 1, we have $\exists r.B \in \pi(\sigma, w)$ and $\pi(\sigma', w) = Q(B)$. Hence, $A \in Q(B)$. Since $\pi(\sigma, w)$ is closed under cl, rule **R6** yields $\exists r.A \in \pi(\sigma, w)$.
 - $\sigma' = \sigma \cdot (P_{=1}r, v, B)$. We apply Claim 2(v) to obtain $\exists P_{=1}r.A \in \pi(\sigma, w)$. Since $\pi(\sigma, w)$ is closed under cl, rule **R4** yields $\exists r.A \in \pi(\sigma, w)$.
“only if”: Let $\exists r.A \in \pi(\sigma, w)$. By rule 1 of the construction, there is a domain element $\sigma' = \sigma \cdot (r, w, A)$ with $\pi(\sigma', w) = Q(A)$, thus $A \in \pi(\sigma', w)$ and, by induction, $\sigma' \in A^{\mathcal{I},w}$. By definition of the interpretation of role names, $(\sigma, \sigma') \in r^{\mathcal{I},w}$. Hence, $\sigma \in (\exists r.A)^{\mathcal{I},w}$.
- $C = \exists P_{=1}r.A$. “if”: Let $\sigma \in (\exists P_{=1}r.A)^{\mathcal{I},w}$, thus there is a domain element σ' with $\sigma' \in A^{\mathcal{I},w}$ and $(\sigma, \sigma') \in r^{\mathcal{I},v}$ for all $v \in W$. By induction hypothesis, $A \in \pi(\sigma', w)$.

Consider now the worlds $0, \varepsilon \in W$: By definition of the interpretation of r , it follows from $(\sigma, \sigma') \in r^{\mathcal{I}, 0} \cap r^{\mathcal{I}, \varepsilon}$ that $\sigma' = \sigma \cdot (P_{=1}r, v, B)$ for some world $v \in W$ and a concept name B . By Claim 2(v), this together with $A \in \pi(\sigma', w)$ yields $\exists P_{=1}r.A \in \pi(\sigma, w)$.

“only if”: Let $\exists P_{=1}r.A \in \pi(\sigma, w)$. By rule 1 of the construction, there is a domain element $\sigma' = \sigma \cdot (P_{=1}r, w, A)$ with $\pi(\sigma', w) = Q(A)$, thus $A \in \pi(\sigma', w)$ and $\sigma' \in A^{\mathcal{I}, w}$. By definition of the interpretation of role names, $(\sigma, \sigma') \in r^{\mathcal{I}, v}$ for all $v \in W$. Hence, $\sigma \in (\exists P_{=1}r.A)^{\mathcal{I}, w}$.

- $C = \exists P_{>0}r.A$. “if”: Let $\sigma \in (\exists P_{>0}r.A)^{\mathcal{I}, w}$, thus there is a $\sigma' \in \Delta^{\mathcal{I}}$ with $\sigma' \in A^{\mathcal{I}, w}$ and $(\sigma, \sigma') \in r^{\mathcal{I}, v}$ for some $v \in W$. By induction hypothesis, $A \in \pi(\sigma', w)$. Again, we distinguish the three cases of the interpretation of the roles.

- $\sigma' = \sigma \cdot (P_{>0}r, v', B)$ and $w = (\sigma, r, B)$ for some concept name B . It follows immediately from Claim 2(v) that $\exists P_{>0}r.A \in \pi(\sigma, w)$.
- $\sigma' = \sigma \cdot (r, w, B)$ for some concept name B . By construction, in particular rule 1, we have $\exists r.B \in \pi(\sigma, w)$ and $\pi(\sigma', w) = Q(B)$. Since $\pi(\sigma, w)$ is closed under cl, rule **R6** implies $\exists r.A \in \pi(\sigma, w)$. Thus, by **R5**, $\exists P_{>0}r.A \in \pi(\sigma, w)$.
- $\sigma' = \sigma \cdot (P_{=1}r, v, B)$. Applying Claim 2(v) yields $\exists P_{=1}r.A \in \pi(\sigma, w)$. Applying rule **R4** we obtain $\exists P_{>0}r.A \in \pi(\sigma, w)$.

“only if”: Let $\exists P_{>0}r.A \in \pi(\sigma, w)$. On the one hand, by rule 1 of the construction there is a domain element $\sigma' = \sigma \cdot (P_{>0}r, w, A)$ with $\pi(\sigma', w) = Q(A)$. By hypothesis we get $\sigma' \in A^{\mathcal{I}, w}$. On the other hand, Claim 2(i) implies $(A_n, A) \in R(P_{>0}r)$. Thus, by rule 3(a), the world $v = (\sigma, r, A)$ exists. By definition of the interpretation of role names $(\sigma, \sigma') \in r^{\mathcal{I}, v}$ for $v = (\sigma, A, r)$. Hence $\sigma \in (\exists P_{>0}r.A)^{\mathcal{I}, w}$.

This finishes the proof of the claim. Using the proved claims it is easy to show that \mathcal{I} is a model of \mathcal{T} . Assume $X_1 \sqcap \dots \sqcap X_n \sqsubseteq X \in \mathcal{T}$ and there are $\sigma \in \Delta^{\mathcal{I}}$ and $w \in W$ such that $\sigma \in X_i^{\mathcal{I}, w}$ for all $1 \leq i \leq n$. By Claim 3, we have $X_i \in \pi(\sigma, w)$ for all $1 \leq i \leq n$. By Claim 2(ii), we know that $\pi(\sigma, w)$ is either $Q(A)$, $Q_{\text{cert}}(A)$, or $\Gamma(t)$ for some trace t to A . In either case $\pi(\sigma, w)$ is closed under the rules cl in Figure 4.4, thus rule **R1** implies $X \in \pi(\sigma, w)$. Another application of Claim 3 yields $\sigma \in X^{\mathcal{I}, w}$.

It remains to show that for $\sigma_0 = (\alpha, \varepsilon, A_0)$ we have $\sigma_0 \in A_0^{\mathcal{I}, \varepsilon}$, but not $\sigma_0 \in B_0^{\mathcal{I}, \varepsilon}$. However, both are obviously true: first we note that, by construction, $\pi(\sigma_0, \varepsilon) = Q(A_0)$. By definition, $A_0 \in Q(A_0)$, hence $\sigma_0 \in A_0^{\mathcal{I}, \varepsilon}$ by the above claim. On the other hand, by assumption we have $B_0 \notin Q(A_0)$, thus by the above claim $\sigma_0 \notin B_0^{\mathcal{I}, \varepsilon}$. ■

We now argue that the algorithm can be implemented using only polynomial space. First, it is easy to see that there can be only polynomially many rule applications: every rule application extends the data structures $Q(\cdot)$, $Q_{\text{cert}}(\cdot)$, and $R(\cdot)$, but these structures

consist of polynomially many sets, each with at most polynomially many elements. It thus remains to verify that each rule application can be executed using only polynomial space. This is obvious for rules **R1-R6** and **S1-S5**. However, for the rules involving traces, i.e., **S6** and **S7**, we have to show that it is not necessary to consider all (infinitely many!) traces. We show the following proposition using a straightforward pumping argument.

Proposition 4.19. *If there is a trace t to B , then there is a trace \hat{t} to B with $\Gamma(\hat{t}) = \Gamma(t)$ and length at most $M := |\mathcal{T}| \cdot 2^{|\mathcal{T}|}$.*

Proof. Let $t = S, A_1, \alpha_2, \dots, \alpha_n, A_n$ and $n > M$. In what follows, Γ_i denotes the type of the trace t_i , i.e., $\Gamma(t_i)$. Consider the sequence $(A_1, \Gamma_1), \dots, (A_n, \Gamma_n)$ of concept names with their corresponding types. Note that there are at most $2^{|\mathcal{T}|}$ possible types and at most $|\mathcal{T}|$ concept names. Since $n > M$, the pigeon hole principle implies that there are $1 \leq i < j \leq n$ with $A_i = A_j$ and $\Gamma_i = \Gamma_j$. It should be clear that the sequence

$$t' = S, A_1, \alpha_2, \dots, \alpha_i, A_i, \alpha_{j+1}, A_{j+1}, \dots, A_n$$

is, in fact, a trace to B and $\Gamma(t') = \Gamma(t)$. Obviously, t' is shorter than t . If the length of t' is at most M , we are done; otherwise, set $t := t'$ and repeat the above steps. ■

Now, to implement **S6** and **S7** in polynomial space, we use a non-deterministic approach, enabled by Savitch's theorem: to check whether there is a trace t to B with $C \in \Gamma(t)$, we guess t step-by-step, at each time keeping only a single A_i, α_i and $\Gamma(t_i)$ in memory. When we reach a situation where $A_i = B$ and $C \in \Gamma(t_i)$, our guessing was successful and we apply the rule. In order to stop this procedure, we also maintain a binary counter of the number of steps that have been guessed so far. As soon as this counter exceeds M from Proposition 4.19, the maximum length of non-repeating traces, we stop the guessing and do not apply the rule. Clearly, this yields an algorithm that runs in polynomial space.

Theorem 4.20. *Subsumption relative to general $\text{Prob}\mathcal{EL}_{01}$ -TBoxes is PSPACE-complete.*

As a byproduct, the proof of Lemma 4.18 yields a unique least model (in the sense of Horn-logic). This can be used in proving convexity of $\text{Prob}\mathcal{EL}_{01}$.

Corollary 4.21. *Subsumption in $\text{Prob}\mathcal{EL}_{01}$ is convex, i.e., $\mathcal{T} \models C \sqsubseteq D_1 \sqcup D_2$ implies $\mathcal{T} \models C \sqsubseteq D_1$ or $\mathcal{T} \models C \sqsubseteq D_2$.*

Proof. We prove the contrapositive. Take an arbitrary general TBox \mathcal{T} and concept names C, D_1 , and D_2 such that

$$\mathcal{T} \not\models C \sqsubseteq D_1 \quad \text{and} \quad \mathcal{T} \not\models C \sqsubseteq D_2.$$

By Lemma 4.18, we get that $D_1 \notin Q(C)$ and $D_2 \notin Q(C)$. Note that the model $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$ constructed in the proof of Lemma 4.18 features some individual $d \in \Delta^{\mathcal{I}}$ and a world w with $d \in C^{\mathcal{I},w}$ and for all concept names: $d \in D^{\mathcal{I},w}$ iff $D \in Q(C)$. Thus, we have $d \in C^{\mathcal{I},w}$ but $d \notin D_1^{\mathcal{I},w}$ and $d \notin D_2^{\mathcal{I},w}$ and hence $d \notin (D_1 \sqcup D_2)^{\mathcal{I},w}$. This proves that $\mathcal{T} \not\models C \sqsubseteq D_1 \sqcup D_2$. \blacksquare

Let us remark that the logic $\text{Prob}\mathcal{EL}_{01}$ is a syntactic variant of the two-dimensional logic $\mathbf{S5}_{\mathcal{EL}}$ which can be seen as a fragment of the description logic ‘of change’ $\mathbf{S5}_{\mathcal{ALCCQI}}$ [6]. More precisely, define C^\dagger to be the $\mathbf{S5}_{\mathcal{EL}}$ concept that is obtained from the $\text{Prob}\mathcal{EL}_{01}$ concept C by replacing $P_{>0}$ with \diamond and $P_{=1}$ with \Box and lift this translation \cdot^\dagger in the natural way to TBoxes. We then have that

$$\mathcal{T} \models C \sqsubseteq D \quad \text{iff} \quad \mathcal{T}^\dagger \models C^\dagger \sqsubseteq D^\dagger.$$

With a similar mapping, we can reduce from subsumption in $\mathbf{S5}_{\mathcal{EL}}$ to subsumption in $\text{Prob}\mathcal{EL}_{01}$. Hence, the PSPACE-completeness result transfers to $\mathbf{S5}_{\mathcal{EL}}$.

Corollary 4.22. *Subsumption in $\mathbf{S5}_{\mathcal{EL}}$ relative to general TBoxes is PSPACE-complete.*

4.4 Conclusion and Outlook

In this chapter, we have taken a closer look at probabilistic extensions of the tractable description logic \mathcal{EL} with the hope of identifying tractable fragments. However, in the first part, we showed that—quite to the contrary—*any* extension of \mathcal{EL} with a single probabilistic operator $P_{\sim p}$ with $p \in (0, 1)$ renders subsumption relative to general TBoxes EXPTIME-hard. Motivated by the fact that many ontologies used in biomedical applications are in fact *classical* TBoxes, we then studied subsumption relative to classical TBoxes. In particular, we were able to identify tractable fragments, namely $\text{Prob}\mathcal{EL}_c^{\sim p}$ for any choice of $\sim \in \{>, \geq\}$. Notably, these fragments coincide for any choice. We complemented this result by showing that for $\sim \in \{<, \leq\}$, these logics are coNP-hard. In the second part, we studied the fragment $\text{Prob}\mathcal{EL}_{01}$ and managed to close the complexity gap to PSPACE-completeness.

Open problems

In the case of probabilistic concepts, classical TBoxes have proved to be a promising alternative to general TBoxes in the sense that they admit polynomial time reasoning for some fragments that are intractable for general TBoxes. We leave as future work the precise complexity for subsumption checking relative to full $\text{Prob}\mathcal{EL}_c$ classical TBoxes, that is, when we allow more than a single probabilistic operator. It would be interesting to also admit probabilistic roles in this framework.

5 Ontology-Based Access to Probabilistic Data

In the introduction, we have argued that applications which require data to be first extracted from the web and then further processed and accessed locally by feeding it into a relational database system (RDBMS) face two crucial difficulties:

- (1) data extracted from the web is often provided without explicit schema information;
- (2) extracted data is often *uncertain* because of the unreliability of many web data sources and due to the data extraction process, which relies on heuristic decisions and is significantly error prone [97].

Item (1) is addressed by the framework of *ontology-based database access (OBDA)*, where an ontology provides background knowledge about the domain and is used for the *interpretation* or *completion* of data. While the current techniques developed in OBDA are well-suited to deal with the first aspect, they are not able to deal with Item (2), uncertainty. Thus, in this chapter,

we assume uncertainty in the data and propose and analyze the framework of *ontology-based access to probabilistic data (pOBDA)*.

In our framework, we adopt data models from the recently very active area of probabilistic databases [32, 125], but use an open world assumption as is standard in the context of OBDA. In a nutshell, our framework pOBDA relates to probabilistic database systems in the same way that traditional OBDA relates to relational database systems. To put it into context with the previous chapters, let us note that we allow for probabilistic data non-probabilistic ontologies formulated in first-order logic or a description logic. In fact, we deliberately avoid probabilities in the ontology because this results in a simple and fundamental, yet useful formalism that still admits a very transparent semantics. Finally, notice that the adopted data model features implicit independence assumptions, which are not made in ProbFO and enable us to encode exponentially many worlds in a succinct way.

While the relevant reasoning problem in traditional OBDA is query answering, we switch here to *computing answer probabilities* to (mostly) conjunctive queries. In database research, practical feasibility is usually identified with PTIME *data complexity*, where data complexity means to treat only the (probabilistic) data as an input while considering both the ontology and the query to be fixed. Hence, the main aim of this chapter is to

study the data complexity of computing answer probabilities in the framework of pOBDA described above.

More precisely, we pursue a non-uniform approach to complexity as recently initiated by [102] and continued in [18]; however, in contrast to the former and similar to the latter, we define one problem for each pair (q, \mathcal{T}) of query and ontology. Notice that computing answer probabilities is *not* a decision problem but rather closely related to counting problems. Thus, instead of identifying hard problems with NP-hardness, we use the natural analog in counting complexity $\#P$ [126].

Our running example is web data extraction where the extracted data is stored in a probabilistic database, in the spirit of [64]. There are plenty of web data extraction tools which often use some kind of confidence score attached to assertions since the extraction process is error prone; two examples are [21, 41]. One particular way to implement these confidence scores is via probabilities. Further note that in many information extraction tools, background knowledge in form of an ontology is already used at the stage of extraction, see for example [51]. This is in contrast to our approach where the ontology is employed during querying. We believe, however, that the two approaches are not excluding each other and can be orchestrated to play together.

Related Work

The probabilistic ABox formalism studied in this chapter is inspired by the probabilistic database models in [36], but can also be viewed as a variation of probabilistic versions of datalog and Prolog, see [116, 50, 107] and references therein. They can also be seen as less succinct version of *pc-tables*, a traditional data model for probabilistic databases due to Imielinski and Lipski [81]. Most relevant for us is the intensive study of tuple-independent databases; in particular, we will exploit the PTIME/ $\#P$ -dichotomy for answering unions of conjunctive queries over such databases [37]. Nowadays, there is an abundance of other probabilistic data models, see [63, 120, 4, 125] and the references therein. All these models provide a compact representation of distributions over potentially large sets of *possible worlds*. Taking into account the open-world assumption and the TBox, our semantics can be compared to probabilistic datalog [116, 49], however, without uncertainty in the TBox.

The motivation for our framework is somewhat similar to what is done in [124], where the retrieval of top- k -answers in OBDA is considered under a fuzzy logic-like semantics based on ‘scoring functions’. There have recently been other approaches to combining ontologies and probabilities for data access [47, 59], yet with a different semantics; the setup considered by Gottlob, Lukasiewicz, and Simari in [59] is close in spirit to the framework studied here, but also allows probabilities in the TBox and has a different, rather intricate semantics based on Markov logic. The proposal by Finger *et al.* [47] differs from our setting as it does not adopt any independence assumptions. There has

also been a large number of proposals for enriching description logic TBoxes (instead of ABoxes) with probabilities, see Chapter 4 and [100, 101] and the references therein. The setting perhaps closest to ours with respect to the semantics is probabilistic data exchange recently introduced by Fagin, Kimelfeld, and Kolaitis [44]. They adopt the same data model, but—as is common for data exchange settings—study the influence of schema mappings including tuple and equality generating dependencies instead of an ontology.

Contribution and Structure of the Chapter

In Section 5.1, we introduce the necessary preliminaries for queries and query answering relative to an ontology in traditional OBDA. In Section 5.2, we introduce the framework of ontology-based access to probabilistic data. First, we formally specify our data model *probabilistic ABoxes* (*pABoxes*) and a restricted variant of it, *assertion-independent probabilistic ABoxes* (*ipABoxes*), which can be viewed as counterpart of tuple-independent databases. Second, we define the relevant computational problem, namely *computing the probability of certain answers* to a query relative to an ontology. Additionally, we observe that allowing full probabilistic ABoxes always leads to #P-hardness and we thus restrict our attention mostly to *ipABoxes*.

As the central tool for studying complexity, we use *query rewriting*, which is an important and well-studied technique for traditional OBDA [28, 94, 17, 90]. In a nutshell, a query q and an ontology \mathcal{T} are rewritten into a new query $q_{\mathcal{T}}$ such that answering q relative to \mathcal{T} is the same as answering $q_{\mathcal{T}}$. The fact that we can use query rewritings from traditional OBDA also in the context of pOBDA is based on the following observation: for any pABox \mathcal{A} , the probability that a tuple \vec{a} is a certain answer to q over \mathcal{A} relative to an ontology \mathcal{T} is identical to the probability that \vec{a} is an answer to $q_{\mathcal{T}}$ over \mathcal{A} viewed as a probabilistic database. This *lifting* of query rewriting to the probabilistic case immediately implies that one can implement pOBDA based on existing PDBMSs such as MayBMS, Trio, and MystiQ [3, 129, 23].

In Section 5.3, we begin our study of the complexity landscape in pOBDA by considering pairs (q, \mathcal{T}) of first-order queries and first-order ontologies which can be rewritten in the above sense. Lifting allows us to carry over the dichotomy between PTIME and #P-hardness for computing the probabilities of answers to unions of conjunctive queries (UCQs) over probabilistic databases recently obtained by Dalvi, Schnaitter, and Suciu [33, 37] to our pOBDA framework provided that we restrict ourselves to *ipABoxes*. That is, each such pair (q, \mathcal{T}) can be answered in polynomial time or it is #P-hard to do so.

In Section 5.4, we instantiate this to concrete ontology and query languages, namely DL-Lite and Boolean, connected conjunctive queries (CQs) without individual names. Most notably, we provide a transparent and decidable characterization of those queries q and DL-Lite-ontologies \mathcal{T} for which computing answer probabilities is in PTIME. As a nec-

essary preliminary step, we restate the mentioned dichotomy of UCQs over probabilistic databases [37].

In Section 5.5, we proceed to showing that query rewriting is a *complete* tool for proving PTIME data complexity in pOBDA, in the following sense: we replace DL-Lite with the strictly more expressive description logic \mathcal{ELI} , where, in contrast to DL-Lite, rewritings into first-order queries do not exist for every CQ q and ontology \mathcal{T} ; we then prove that if any (q, \mathcal{T}) does *not* have a rewriting, then computing answer probabilities for q relative to \mathcal{T} is $\#P$ -hard. Thus, if it is possible at all to prove that some (q, \mathcal{T}) has PTIME data complexity, then this can always be done using query rewriting.

Both in DL-Lite and \mathcal{ELI} , the class of queries and TBoxes with PTIME data complexity is relatively small, which leads us to also consider the approximation of answer probabilities, based on the notion of a *fully polynomial randomized approximation scheme* (FPRAS). This is the subject of Section 5.6. It is not hard to see that all pairs (q, \mathcal{T}) have an FPRAS when \mathcal{T} is formulated in DL-Lite. Even better, this result generalizes to a more expressive data model which allows for DNF annotations. This observation clearly gives hope for practical applications. As in the exact, case we move to the ontology language \mathcal{ELI} and show that FO-rewritability is again the right tool to study (non-)existence of FPRASes. Our two main results are as follows. Over ipABoxes, we choose one non-FO-rewritable pair (q, \mathcal{T}) and show that there is an FPRAS if, and only if there is an FPRAS for a certain notoriously hard probabilistic network reliability problem. Over pABoxes allowing for DNF annotations, we show that the existence of an FPRAS for a Boolean, connected q and \mathcal{ELI} -TBox \mathcal{T} is equivalent to FO-rewritability of q relative to \mathcal{T} .

5.1 Preliminaries

A *first-order query* (FOQ) is a first-order formula $\varphi(\vec{x})$ constructed from atoms $A(t)$ and $r(t, t')$ using negation, conjunction, and existential quantification where t, t' denote *terms*, that is, variable symbols or individual names. The free variables \vec{x} are the *answer variables* of $\varphi(\vec{x})$. A FOQ φ is *n-ary* if it has n answer variables and *Boolean* if it is 0-ary. We will mostly consider a special class of FOQs: *conjunctive queries* (CQs) take the form $\exists \vec{y} \varphi(\vec{x}, \vec{y})$, where φ a conjunction of atoms of the form $A(t)$ and $r(t, t')$. We call the variables in \vec{y} the *quantified* variables. The set of all variables in a CQ q is denoted by $\text{var}(q)$ and the set of all terms in q by $\text{term}(q)$. Whenever convenient, we treat a CQ as a *set* of atoms and sometimes write $r^-(t, t')$ instead of the atom $r(t', t)$. *Unions of conjunctive queries* (UCQs) are disjunctions of CQs. A conjunctive query q is called *connected* if the graph (V_q, E_q) is connected, where

- $V_q = \text{term}(q)$;
- $E_q = \{\{t, t'\} \mid r(t, t') \in q\}$.

Let φ be an n -ary FOQ. For an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})^1$, we write $\text{ans}(\varphi, \mathcal{I})$ to denote the *answers to φ in \mathcal{I}* , that is, the set of all tuples $\vec{a} \in (\Delta^{\mathcal{I}})^n$ such that $\mathcal{I} \models \varphi[\vec{a}]$. For conjunctive queries, answers are characterized using the notion of matches. More specifically, let $q(\vec{x})$ be a CQ with answer variables (x_1, \dots, x_n) . For $\vec{a} = a_1 \cdots a_n \in (\mathbf{N}_I)^n$, an \vec{a} -match for q in \mathcal{I} is a mapping $\pi : \text{term}(q) \rightarrow \Delta^{\mathcal{I}}$ such that:

- $\pi(x_i) = a_i$ for $1 \leq i \leq n$,
- $\pi(a) = a^{\mathcal{I}}$ for all $a \in \text{term}(q) \cap \mathbf{N}_I$, and
- $\pi(t) \in A^{\mathcal{I}}$ for all $A(t) \in q$, and $(\pi(t_1), \pi(t_2)) \in r^{\mathcal{I}}$ for all $r(t_1, t_2) \in q$.

Obviously, \vec{a} is an answer to q in \mathcal{I} iff there is an \vec{a} -match of q in \mathcal{I} . We extend the notion of matches to UCQs by saying that a UCQ q has an \vec{a} -match in \mathcal{I} if one disjunct of q has an \vec{a} -match in \mathcal{I} .

We use the formal term *TBox* instead of “ontology”, and let a TBox just be a finite set of first-order sentences. For a TBox \mathcal{T} , an ABox \mathcal{A} , and a FOQ $\varphi(\vec{x})$ we write $\mathcal{T}, \mathcal{A} \models \varphi[\vec{a}]$ if $\mathcal{I} \models \varphi[\vec{a}]$ for all models \mathcal{I} of \mathcal{T} and \mathcal{A} . In this case and when all elements of \vec{a} are from $\text{Ind}(\mathcal{A})$, \vec{a} is a *certain answer to φ w.r.t. \mathcal{A} and \mathcal{T}* . We use $\text{cert}_{\mathcal{T}}(\varphi, \mathcal{A})$ to denote the set of all certain answers to φ w.r.t. \mathcal{A} and \mathcal{T} .

Ontology-based data access (OBDA) is the problem of finding the certain answers. More precisely, the corresponding decision problem is as follows:

CERTAIN ANSWER

INPUT: TBox \mathcal{T} , ABox \mathcal{A} , query $\varphi(\vec{x})$, candidate answer \vec{a}

OUTPUT: Is $\vec{a} \in \text{cert}_{\mathcal{T}}(\varphi, \mathcal{A})$?

Thus, in this setting, one adopts the *open-world assumption*: the data (stored in the ABox) is assumed to be incomplete and the ontology \mathcal{T} is used to infer implicit knowledge. In general, finding certain answers requires looking at all models of \mathcal{A} and \mathcal{T} which is in contrast to just computing answers without an ontology; intuitively, we move from model checking to inference problems.

For two CQs q, q' , we write $q \sqsubseteq q'$ in case $\text{ans}(q, \mathcal{I}) \subseteq \text{ans}(q', \mathcal{I})$ for all interpretations \mathcal{I} , that is, when q *implies* q' . We say that a CQ q is *minimal* if there is no strict sub-query $q' \subsetneq q$ with $q' \sqsubseteq q$. Implication among Boolean conjunctive queries can be conveniently characterized via the notion of homomorphisms. A *homomorphism from q' to q* is a mapping $h : \text{term}(q') \rightarrow \text{term}(q)$ such that:

- $h(a) = a$ for each $a \in \text{term}(q) \cap \mathbf{N}_I$;
- $A(h(t)) \in q$ for all $A(t) \in q'$, and $r(h(t_1), h(t_2)) \in q$ for all $r(t_1, t_2) \in q'$.

¹Note that, for the sake of uniformity, we use a notation that is typically used for DLs, since we will mainly consider DLs as ontology language.

It is well-known that $q \sqsubseteq q'$ iff there is a homomorphism from q' to q . We say that a homomorphism from q' to q is *atom-surjective* if for every atom $A(t) \in q$ (respectively, $r(t_1, t_2) \in q$), there is some atom $A(t') \in q'$ (resp., $r(t'_1, t'_2) \in q'$) with $h(t') = t$ (resp., $h(t'_1) = t_1$ and $h(t'_2) = t_2$). Under this definition, q is minimal iff there are only atom-surjective homomorphisms from q to itself.

Often, we will view queries as ABoxes, and ABoxes as interpretations. In particular, given a CQ q without individual names, introduce an ABox individual a_x for each variable $x \in \text{var}(q)$ and define \mathcal{A}_q as the set of all assertions $\{A(a_x) \mid A(x) \in q\} \cup \{r(a_x, a_y) \mid r(x, y) \in q\}$. For an ABox \mathcal{A} , define the interpretation $\mathcal{I}_{\mathcal{A}} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ by taking

- $\Delta^{\mathcal{I}} = \text{Ind}(\mathcal{A})$;
- $A^{\mathcal{I}} = \{a \mid A(a) \in \mathcal{A}\}$ for all concept names A ; and
- $r^{\mathcal{I}} = \{(a, b) \mid r(a, b) \in \mathcal{A}\}$ for all role names r .

This gives rise to an interpretation $\mathcal{I}_q = \mathcal{I}_{\mathcal{A}_q}$ for each query q . Note that for Boolean CQs, there is a match of q in $\mathcal{I}_{q'}$ iff there is a homomorphism from q to q' iff $q' \sqsubseteq q$. Since for Boolean CQs the conditions for having a match are very similar to the homomorphism conditions, we will sometimes say that there is a homomorphism (instead of a match) from a query into an interpretation.

As done often in the context of OBDA, we adopt the unique name assumption (UNA), which requires that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all interpretations \mathcal{I} and all $a, b \in \mathbb{N}_I$ with $a \neq b$. This has no influence on the complexity results obtained.

5.2 The Framework of Probabilistic OBDA

In this section, we introduce our framework of *ontology-based access to probabilistic data (probabilistic OBDA)*. Let us start with defining the data model underlying our approach. It is a rather general, probabilistic version of ABoxes which can be viewed as an open-world variant of probabilistic databases in the sense of [36].

Let \mathcal{E} be a countably infinite set of *atomic (probabilistic) events*. An *event expression* is built up from atomic events using the Boolean operators \neg, \wedge, \vee . We use $\text{expr}(\mathcal{E})$ to denote the set of all event expressions over \mathcal{E} . A *probability assignment for E* is a map $E \rightarrow [0, 1]$.

Definition 5.1 (pABox). A probabilistic ABox (pABox) is of the form (\mathcal{A}, e, p) with \mathcal{A} an ABox, e a map $\mathcal{A} \rightarrow \text{expr}(\mathcal{E})$, and p a probability assignment for $E_{\mathcal{A}}$, the atomic events in \mathcal{A} .

We consider as a running example a (fictitious) information extraction tool that is gathering data from the web, see [64] for a similar setup.

Example 5.2. Assume we are gathering data about soccer players and the clubs they play for in the current 2014 season, and we want to represent the result as a pABox.

(1) The tool processes a newspaper article stating that ‘Messi is the soul of the Argentinian national soccer team’. Because the exact meaning of this phrase is unclear (it could refer to a soccer player, a coach, a mascot), it generates the assertion `Player(messi)` associated with the atomic event expression e_1 with $p(e_1) = 0.7$. The event e_1 represents that the phrase was interpreted correctly.

(2) The tool finds the Wikipedia page on Lionel Messi, which states that he is a soccer player. Since Wikipedia is typically reliable and up to date, but not always correct, it updates the expression associated with `Player(messi)` to $e_1 \vee e_2$ and associates e_2 with $p(e_2) = 0.95$.

(3) The tool finds an HTML table on the homepage of FC Barcelona saying that the top scorers of the season are Messi, Villa, and Pedro. It is not stated whether the table refers to the 2013 or the 2014 season, and consequently we generate the ABox assertions `playsfor(x , FCbarca)` for $x \in \{\text{messi, villa, pedro}\}$ all associated with the same atomic event expression e_3 with $p(e_3) = 0.5$. Intuitively, the event e_3 is that the table refers to 2014.

(4) Still processing the table, the tool applies the background knowledge that top scorers are typically strikers. It generates three assertions `Striker(x)` with $x \in \{\text{messi, villa, pedro}\}$, associated with atomic events e_4 , e'_4 , and e''_4 . It sets $p(e_4) = p(e'_4) = p(e''_4) = 0.8$. The probability is higher than in (3) since being a striker is a more stable property than playing for a certain club, thus this information does not depend so much on whether the table is from 2013 or 2014.

(5) The tool processes the twitter message ‘Villa was the only one to score a goal in the match between Barca and Real’. It infers that Villa plays either for Barcelona or for Madrid, generating the assertions `playsfor(villa, FCbarca)` and `playsfor(villa, realmartid)`. The first assertion is associated with the event e_5 , the second one with $\neg e_5$. It sets $p(e_5) = 0.5$.

Intuitively, probabilistic ABoxes encode in a succinct way a distribution over a set of possible worlds much in the way as probabilistic databases: every possible truth assignment to the atomic events corresponds to a world whose probability is the probability of the truth assignment (note: the atomic events are assumed independent). However, as usual for OBDA, we adopt the open-world assumption, hence probabilistic ABoxes can be viewed to encode a distribution over a set of possible *open* worlds. In our framework of probabilistic OBDA, one intuitively computes the certain answers in every such world and weights them according to the probability of the world. To be more precise, we introduce the semantics of pABoxes (\mathcal{A}, e, p) and the probability of certain answers. As query language, we choose first-order queries since they are most general; later, we will often work with conjunctive queries. Note that each $E \subseteq E_{\mathcal{A}}$ can be viewed as a truth assignment that makes all events in E true and all events in $E_{\mathcal{A}} \setminus E$ false; we write

$E \models \psi$ in case a propositional formula ψ evaluates to 1 under E (viewed as a truth assignment).

Definition 5.3 (Semantics). Let (\mathcal{A}, e, p) be a $pABox$. For each $E \subseteq E_{\mathcal{A}}$, define a corresponding non-probabilistic $ABox$ $\mathcal{A}_E := \{\alpha \in \mathcal{A} \mid E \models e(\alpha)\}$. The function p represents a probability distribution on $2^{E_{\mathcal{A}}}$, by setting for each $E \subseteq E_{\mathcal{A}}$:

$$p(E) = \prod_{e \in E} p(e) \cdot \prod_{e \in E_{\mathcal{A}} \setminus E} (1 - p(e)).$$

The probability of an answer $\vec{a} \in \text{Ind}(\mathcal{A})^n$ to an n -ary first-order query $\varphi(\vec{x})$ over a $pABox$ \mathcal{A} and $TBox$ \mathcal{T} is

$$p_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi) = \sum_{\substack{E \subseteq E_{\mathcal{A}}, \\ \vec{a} \in \text{cert}_{\mathcal{T}}(\varphi, \mathcal{A}_E)}} p(E).$$

For Boolean queries φ , we write $p(\mathcal{T}, \mathcal{A} \models \varphi)$ instead of $p_{\mathcal{T}, \mathcal{A}}(() \in \varphi)$, where $()$ denotes the empty tuple.

Thus, for every $E \subseteq E_{\mathcal{A}}$, the $ABox$ \mathcal{A}_E refers to a possible world with probability $p(E)$. Moreover, we assume pairwise independence of the atomic probabilistic events; this is reflected in the computation of $p(E)$. We pick up the web data extraction example discussed above and illustrate how ontologies can help to reduce uncertainty.

Example 5.4. In the example, we use the *DL-Lite* $TBox$

$$\mathcal{T} = \{ \begin{array}{ll} \exists \text{playsfor} \sqsubseteq \text{Player} & \text{Player} \sqsubseteq \exists \text{playsfor} \\ \exists \text{playsfor}^- \sqsubseteq \text{SoccerClub} & \text{Striker} \sqsubseteq \text{Player} \end{array} \}$$

and consider the following subcases of the example above.

(1) + (3) The resulting $pABox$ comprises the following assertions with associated event expressions:

$$\begin{array}{ll} \text{Player}(\text{messi}) \rightsquigarrow e_1 & \text{playsfor}(\text{messi}, \text{FCbarca}) \rightsquigarrow e_3 \\ \text{playsfor}(\text{villa}, \text{FCbarca}) \rightsquigarrow e_3 & \text{playsfor}(\text{pedro}, \text{FCbarca}) \rightsquigarrow e_3 \end{array}$$

with $p(e_1) = 0.7$ and $p(e_3) = 0.5$. Without a $TBox$, *messi* is an answer to the query $\text{Player}(x)$ with probability 0.7, independent of the event e_3 . Because of the statement $\exists \text{playsfor} \sqsubseteq \text{Player}$, using \mathcal{T} (instead of the empty $TBox$) increases the probability of *messi* to be an answer to the query $\text{Player}(x)$ from 0.7 to 0.85: there is only one world \mathcal{A}_E where *messi* is not certainly a player, namely for $E = \emptyset$ whose probability is 0.15.

(5) The resulting $pABox$ is

$$\text{playsfor}(\text{villa}, \text{FCbarca}) \rightsquigarrow e_5 \quad \text{playsfor}(\text{villa}, \text{realmadrid}) \rightsquigarrow \neg e_5$$

with $p(e_5) = 0.5$. Although $\text{Player}(\text{villa})$ does not occur in the data, villa is an instance of Player in every possible world, again by the TBox-statement $\exists \text{playsfor} \sqsubseteq \text{Player}$. Thus, the probability of villa to be an answer to the query $\text{Player}(x)$ is 1.

(3)+(4) This results in the pABox

$$\begin{aligned} \text{playsfor}(\text{messi}, \text{FCbarca}) &\rightsquigarrow e_3 & \text{Striker}(\text{messi}) &\rightsquigarrow e_4 \\ \text{playsfor}(\text{villa}, \text{FCbarca}) &\rightsquigarrow e_3 & \text{Striker}(\text{villa}) &\rightsquigarrow e'_4 \\ \text{playsfor}(\text{pedro}, \text{FCbarca}) &\rightsquigarrow e_3 & \text{Striker}(\text{pedro}) &\rightsquigarrow e''_4 \end{aligned}$$

with $p(e_3) = 0.5$ and $p(e_4) = p(e'_4) = p(e''_4) = 0.8$. Due to the last three CIs in \mathcal{T} , each of messi , villa , pedro is an answer to the CQ $\exists y \text{playsfor}(x, y) \wedge \text{SoccerClub}(y)$ with probability 0.9.

Of course, some of the ABoxes \mathcal{A}_E might be inconsistent w.r.t. the TBox \mathcal{T} used. In this case, it may be undesirable to let them contribute to the probabilities of answers. For example, if we use the pABox

$$\text{Striker}(\text{messi}) \rightsquigarrow e_1 \quad \text{Goalie}(\text{messi}) \rightsquigarrow e_2$$

with $p(e_1) = 0.8$ and $p(e_2) = 0.3$ and the TBox $\text{Goalie} \sqcap \text{Striker} \sqsubseteq \perp$, then messi is an answer to the query $\text{SoccerClub}(x)$ with probability 0.24 while one would probably expect it to be zero (which is the result when the empty TBox is used). We follow Antova, Koch, and Olteanu and advocate a pragmatic solution based on *rescaling* [4]. More specifically, we remove those ABoxes \mathcal{A}_E that are inconsistent w.r.t. \mathcal{T} and rescale the remaining set of ABoxes so that they sum up to probability one. In other words, we set

$$\widehat{p}_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi) = \frac{p_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi) - p(\mathcal{T}, \mathcal{A} \models \perp)}{1 - p(\mathcal{T}, \mathcal{A} \models \perp)}$$

where \perp is a Boolean query that is entailed exactly by those ABoxes \mathcal{A} that are inconsistent w.r.t. \mathcal{T} . The rescaled probability $\widehat{p}_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi)$ can be computed in PTIME when this is the case both for $p_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi)$ and $p(\mathcal{T}, \mathcal{A} \models \perp)$. Note that rescaling results in some effects that might be unexpected such as reducing the probability of messi to be an answer to $\text{Striker}(x)$ from 0.8 to ≈ 0.74 when the above TBox is added. However, increased uncertainty about messi being a Striker is not surprising since the TBox and ABox are contradictory, even if the independence assumption is dropped. In the remainder of the chapter, for simplicity we will only admit TBoxes \mathcal{T} such that all ABoxes \mathcal{A} are consistent w.r.t. \mathcal{T} .

5.2.1 Computational Problems

The main computational problem in traditional OBDA is, given an ABox \mathcal{A} , query φ , and TBox \mathcal{T} , to produce the certain answers of φ relative to \mathcal{A} and \mathcal{T} . In our framework

of *probabilistic* OBDA, we rather want to compute the probabilities $p_{\mathcal{T},\mathcal{A}}(\vec{a} \in \varphi)$ of certain answers. More precisely, throughout this chapter we will study the following family of problems indexed by fixed first-order queries φ and TBoxes \mathcal{T} .

pOBDA(φ, \mathcal{T})

INPUT: pABox \mathcal{A} , candidate answer \vec{a}

OUTPUT: answer probability $p_{\mathcal{T},\mathcal{A}}(\vec{a} \in \varphi)$

Thus, as recently initiated in [102], we pursue a *non-uniform approach* to study *data complexity*. The notion of data complexity was introduced by Vardi [127] based on the assumption that queries are typically small compared to data. Given the big amounts of data available and to manage today, we argue that this is also the right complexity measure for our setting. Note, though, that our framework yields one problem for each query and TBox, while [102] has one problem for each TBox, that is, the query is part of the input, similar to the setting in [18]. This is justified by the fact that there are virtually no TBoxes such that all queries are tractable. Ideally, we would like to understand the precise complexity of every query φ relative to every TBox \mathcal{T} , against the background of some preferably expressive ‘master logic’ used for \mathcal{T} .

When considering data complexity for decision problems, one typically identifies easy instances with PTIME while hard instances are characterized by NP-hardness. As pOBDA(φ, \mathcal{T}) is not a decision problem, but rather closely related to counting problems, we use #P as the natural analog to NP in counting complexity to identify intractable problems. In particular, we say:

- a query φ is in PTIME relative to a TBox \mathcal{T} , or (with a slight abuse of notation) $\text{pOBDA}(\varphi, \mathcal{T}) \in \text{PTIME}$, if there is a polynomial time algorithm that, given an ABox \mathcal{A} and a candidate answer $\vec{a} \in \text{Ind}(\mathcal{A})^n$ to φ , computes $p_{\mathcal{A},\mathcal{T}}(\vec{a} \in \varphi)$;
- a query φ is #P-hard relative to \mathcal{T} if the aforementioned problem $\text{pOBDA}(\varphi, \mathcal{T})$ is hard for the counting complexity class #P.

As it is central to this chapter we give some details and complete problems for the class #P, essentially following the seminal paper by Valiant [126]. Intuitively, #P is the ‘counting equivalent’ of NP and consists of all functions $f : \Sigma^* \rightarrow \mathbb{N}$ such that there is a polynomial time, non-deterministic Turing machine M_f whose computation tree on input w has precisely $f(w)$ accepting configurations for all $w \in \Sigma^*$. We define #P-hardness via Turing reductions, that is, reductions that have access to an oracle. In particular, we say that a function f is #P-hard if every function $g \in \#P$ can be computed by a polynomial time algorithm that uses f as an oracle. The first problem shown to be #P-complete is the problem #SAT, that is, given a propositional formula, count the number of satisfying assignments. Remarkably, it has turned out that a counting problem is often much more difficult than its corresponding decision problem. For this chapter, the most relevant

such example is $\#MonBiDNF$ which is the problem of counting the number of satisfying assignments to *monotone bipartite DNF formulas*, that is, formulas of the form

$$(x_{i_1} \wedge y_{i_1}) \vee \dots \vee (x_{i_k} \wedge y_{i_k})$$

such that the sets $\{x_{i_1}, \dots, x_{i_k}\}$ and $\{y_{i_1}, \dots, y_{i_k}\}$ are disjoint. Note that such every formula is trivially satisfiable. In contrast, Provan and Ball proved the following surprising theorem, which has turned out to be a useful tool to show $\#P$ -lower bounds in related settings, see for example [36, 37, 61].

Theorem 5.5 ([113]). *The problem $\#MonBiDNF$ is $\#P$ -hard.*

Of particular interest when studying non-uniform complexity are *dichotomy theorems*. Dichotomy theorems have been of interest for a long time in the areas of constraint satisfaction problems, probabilistic databases, and more recently also in traditional OBDA, see further [45, 102, 37, 25, 122]. In the context of our framework, dichotomy theorems for a fixed class \mathcal{C} of pairs (φ, \mathcal{T}) of queries and TBoxes read as follows:

For every $(\varphi, \mathcal{T}) \in \mathcal{C}$, $\text{pOBDA}(\varphi, \mathcal{T})$ is either in PTIME or $\#P$ -hard.

Naturally, such theorems should speak over preferably rich classes \mathcal{C} . Dichotomy results are interesting, because they indicate a good understanding of the class of problems given by \mathcal{C} . However, a dichotomy theorem in the above form might be *abstract* in the sense that it does not provide a transparent characterization, that is, it does not tell *which* pairs (φ, \mathcal{T}) are tractable and which are not. We will encounter later both abstract theorems and dichotomies containing such characterizations.

5.2.2 Assertion-independent probabilistic ABoxes

We start with the observation that, unsurprisingly, pABoxes are too strong a formalism to admit *any* useful tractable queries. Call a Boolean conjunctive query q *trivial* for \mathcal{T} if $\mathcal{T} \models q$.

Theorem 5.6. *For conjunctive queries q , we have:*

- $\text{pOBDA}(q, \mathcal{T})$ is $\#P$ -hard for every non-Boolean CQ q and TBox \mathcal{T} ;
- $\text{pOBDA}(q, \mathcal{T})$ is $\#P$ -hard for every Boolean CQ q and TBox \mathcal{T} for which it is not trivial. If q is trivial for \mathcal{T} , then $p_{\mathcal{T}, \mathcal{A}}(q) = 1$ for all \mathcal{A} .

Proof. The proof is by reduction of counting the number of satisfying assignments of a propositional formula. Assume that q has answer variables x_1, \dots, x_n and let φ be a propositional formula over variables z_1, \dots, z_m . Convert q, φ into a pABox \mathcal{A} as follows: take q viewed as an ABox, replacing every variable x with an individual name a_x ; then associate every ABox assertion with φ viewed as an event expression over events

z_1, \dots, z_m and set $p(z_i) = 0.5$ for all i . We are interested in the answer $\vec{a} = a_{x_1} \cdots a_{x_n}$. For all $E \subseteq E_{\mathcal{A}}$ with $E \not\models \varphi$, we have $\mathcal{A}_E = \emptyset$. For a non-Boolean CQ q , we have $\vec{a} \notin \text{cert}_{\mathcal{T}}(q, \mathcal{A}_E)$ since $\text{Ind}(\mathcal{A}_E) = \emptyset$; for a Boolean CQ q , we have $() \notin \text{cert}_{\mathcal{T}}(q, \emptyset)$, since $\mathcal{T} \not\models q$. For all $E \subseteq E_{\mathcal{A}}$ with $E \models \varphi$, the ABox \mathcal{A}_E is the ABox-representation of q and thus $\vec{a} \in \text{cert}_{\mathcal{T}}(q, \mathcal{A}_E)$. Consequently, the number of assignments that satisfy φ is $p_{\mathcal{A}, \mathcal{T}}(\vec{a} \in q) \cdot 2^m$. Thus, there is a PTIME algorithm for counting the number of satisfying assignments given an oracle for computing answer probabilities for q and \mathcal{T} . ■

Observe that the reduction relies solely on the expressiveness of the associated event expressions. In particular, restricting the annotations in pABoxes to DNF formulas does not amend this strong negative result as also #DNF is #P-hard. Hence, Theorem 5.6 motivates the study of more lightweight probabilistic ABox formalisms. While pABoxes roughly correspond to *pc-tables*, which are among the most expressive probabilistic data models, we now move to the other end of the spectrum and introduce ipABoxes as a counterpart of *tuple-independent databases* [36, 50]. From a pragmatic perspective, the latter are arguably the most inexpressive probabilistic data model that is still useful, see [125] for more discussion.

Definition 5.7 (ipABoxes). *An assertion-independent probabilistic ABox (ipABox) is a probabilistic ABox in which all event expressions are atomic and where each atomic event expression is associated with at most one ABox assertion.*

To save notation, we write ipABoxes in the form (\mathcal{A}, p) where \mathcal{A} is an ABox and p is a map $\mathcal{A} \rightarrow [0, 1]$ that assigns a probability to each ABox assertion. In this representation, the events are only implicit (one atomic event per ABox assertion). We adapt Definition 5.3 for ipABoxes by defining $p(\mathcal{A}')$ for each $\mathcal{A}' \subseteq \mathcal{A}$ as

$$p(\mathcal{A}') = \prod_{\alpha \in \mathcal{A}'} p(\alpha) \cdot \prod_{\alpha \in \mathcal{A} \setminus \mathcal{A}'} (1 - p(\alpha)).$$

Thus, all assertions in the ABox \mathcal{A} are viewed as independent events. Accordingly, for first-order queries φ , we define

$$p_{\mathcal{A}, \mathcal{T}}(\vec{a} \in \varphi) = \sum_{\substack{\mathcal{A}' \subseteq \mathcal{A} \\ \vec{a} \in \text{cert}_{\mathcal{T}}(\varphi, \mathcal{A}')}} p(\mathcal{A}').$$

Reconsidering Example 5.2, observe that cases (1) and (4) yield ipABoxes, whereas cases (2), (3), and (5) do not. For the remainder of the chapter, we assume that only ipABoxes are admitted unless explicitly noted otherwise. Particularly, we modify the problem $\text{pOBDA}(\varphi, \mathcal{T})$ to admit only ipABoxes as input and define:

$\text{ipOBDA}(\varphi, \mathcal{T})$

INPUT: ipABox \mathcal{A} , candidate answer \vec{a}

OUTPUT: answer probability $p_{\mathcal{A}, \mathcal{T}}(\vec{a} \in \varphi)$

We have already mentioned that our pABoxes can be seen as open world versions of probabilistic databases in the sense of [36]. In the same way, assertion-independent pABoxes are the counterpart of *tuple-independent probabilistic databases* (restricted to relations of arity at most 2). A tuple-independent probabilistic database is a tuple (R_1, \dots, R_k, P) such that

- (R_1, \dots, R_k) is a classical database, that is, each R_i is a relation;
- P is a function assigning each tuple appearing in some R_i a probability.

For example, for an ipABox (\mathcal{A}, p) , we can define a corresponding tuple-independent database as follows. Every assertion $\alpha = A(a)$ with probability $p(\alpha)$ corresponds to a tuple (a) in relation A and $P(a \in A) = p(\alpha)$ is the attached probability; analogously, a role assertion $\alpha = r(a, b)$ with probability $p(\alpha)$ corresponds to the tuple (a, b) in a relation R with assigned probability $P((a, b) \in R) = p(\alpha)$. We use $p_{\mathcal{A}}^d(\vec{a} \in \varphi)$ to denote the probability that \vec{a} is an answer to the query φ given \mathcal{A} *viewed as a tuple-independent probabilistic database in the described way*. A closer look at the semantics of probabilistic databases yields that in fact it agrees with our semantics given an empty TBox.

Observation 5.8. *For each ipABox \mathcal{A} , first-order query φ , and possible answer \vec{a} we have $p_{\mathcal{A}}^d(\vec{a} \in \varphi) = p_{\mathcal{A}, \emptyset}(\vec{a} \in \varphi)$.*

The following dichotomy theorem was recently shown in a series of papers [36, 35, 33, 37].

Theorem 5.9. *For every fixed UCQ q , computing the probability $p_{\mathcal{A}}^d(\vec{a} \in q)$ on input \mathcal{A}, \vec{a} is either in PTIME or #P-hard.*

By Observation 5.8, we can state it in terms of our notation.

Corollary 5.10. *For a fixed UCQ q , the problem $\text{ipOBDA}(q, \emptyset)$ is either in PTIME or #P-hard.*

5.3 The Dichotomy for First-Order Rewritable (φ, \mathcal{T})

We embark on our study of the complexity landscape of $\text{ipOBDA}(\varphi, \mathcal{T})$ by first considering *query rewritings*, an important and well-studied tool for traditional OBDA. The goal of this section is to show that traditional query rewritings crystallize to have immense practical and theoretical consequences in our probabilistic framework. In fact, we will show the first, and most general dichotomy result. In order to proceed to the result, we first introduce the necessary notions.

Definition 5.11 (FO-rewritings). A FOQ $\varphi(\vec{x})$ is FO-rewritable relative to a TBox \mathcal{T} if one can effectively construct a FOQ $\varphi_{\mathcal{T}}(\vec{x})$ such that $\text{cert}_{\mathcal{T}}(\varphi, \mathcal{A}) = \text{ans}(\varphi_{\mathcal{T}}, \mathcal{I}_{\mathcal{A}})$ for every ABox \mathcal{A} . In this case, $\varphi_{\mathcal{T}}(\vec{x})$ is called a rewriting of φ relative to \mathcal{T} . If $\varphi_{\mathcal{T}}(\vec{x})$ is a UCQ, it is called UCQ-rewriting.

The importance of FO-rewritings in traditional OBDA is immediate from this definition and the fact that database management systems are highly optimized for the task of FOQ answering: for computing the certain answers to φ relative \mathcal{A} and \mathcal{T} , one can simply construct a first-order rewriting $\varphi_{\mathcal{T}}$ and then hand it over for execution to a database system that stores \mathcal{A} (viewed as database). Note that, in view of data complexity, the actual size of $\varphi_{\mathcal{T}}$ does not matter. The following theorem demonstrates the effects of FO-rewritability in our framework.

Theorem 5.12. For every FOQ φ and TBox \mathcal{T} such that φ is first-order rewritable relative to \mathcal{T} , $\text{ipOBDA}(\varphi, \mathcal{T})$ is either in PTIME or it is #P-hard.

The proof of Theorem 5.12 involves the application of two deep theorems, namely Theorem 5.9 and Rossman’s homomorphism preservation theorem. We start with observing that first-order rewritings from traditional OBDA are also useful in our framework of probabilistic OBDA. The following ‘lifting theorem’ is immediate from the definitions.

Theorem 5.13 (Lifting Theorem). Let \mathcal{T} be an FO-TBox, \mathcal{A} a pABox, $\varphi(\vec{x})$ an n -ary FOQ, $\vec{a} \in \text{Ind}(\mathcal{A})^n$ a candidate answer for q , and $\varphi_{\mathcal{T}}(\vec{x})$ an FO-rewriting of φ relative to \mathcal{T} . Then we have $p_{\mathcal{T}, \mathcal{A}}(\vec{a} \in \varphi) = p_{\emptyset, \mathcal{A}}(\vec{a} \in \varphi_{\mathcal{T}})$.

Proof. We have:

$$\begin{aligned} p_{\mathcal{A}, \mathcal{T}}(\vec{a} \in q) &= \sum_{E \subseteq E_{\mathcal{A}} \mid \vec{a} \in \text{cert}_{\mathcal{T}}(q, \mathcal{A}_E)} p(E) \\ &= \sum_{E \subseteq E_{\mathcal{A}} \mid \vec{a} \in \text{ans}(\varphi, \mathcal{A})} p(E) \\ &= p_{\emptyset, \mathcal{A}}(\vec{a} \in \varphi). \end{aligned}$$

■

Theorem 5.13 is interesting from an application perspective as it enables the use of probabilistic database systems such as MayBMS, Trio, and MystiQ for implementing probabilistic OBDA [3, 129, 23]. In analogy to traditional OBDA, we compute the FO-rewriting φ of q and \mathcal{T} and feed it to the probabilistic database system. Although it might be necessary to adapt pABoxes in an appropriate way in order to match the data models of these systems, such modifications do not impair applicability of Theorem 5.13.

From a theoretical viewpoint, Theorem 5.13 establishes query rewriting as a useful tool for analyzing data complexity in probabilistic OBDA. In fact, Theorem 5.13 implies that, with respect to data complexity, computing the query probability of φ relative to \mathcal{A} and \mathcal{T} is the same problem as computing the query probability of $\varphi_{\mathcal{T}}$ in \mathcal{A} .

Corollary 5.14. *If $\varphi_{\mathcal{T}}$ is an FO-rewriting of φ relative to \mathcal{T} , then $\text{ipOBDA}(\varphi, \mathcal{T}) = \text{ipOBDA}(\varphi_{\mathcal{T}}, \emptyset)$.*

Next, we show that whenever there is a first-order rewriting for a query relative to some TBox, then there is a UCQ-rewriting. This generalizes a proposition given in [17] for atomic queries, that is, queries of the form $A(x)$, and \mathcal{ELI} -TBoxes to our setting of FO-queries and FO-TBoxes and uses Rossman's homomorphism preservation theorem [119].

Proposition 5.15. *For every FOQ φ and FO-TBox \mathcal{T} such that φ is FO-rewritable relative to \mathcal{T} , we can effectively construct a UCQ-rewriting $\varphi_{\mathcal{T}}$ of φ relative to \mathcal{T} .*

Proof. Let $\widehat{\varphi}(\vec{x})$ be a FOQ with answer variables $\vec{x} = (x_1, \dots, x_k)$, \mathcal{T} an FO-TBox, and $\varphi(\vec{x})$ their FO-rewriting. The proof strategy is to first show that there is a FO-rewriting $\varphi'(\vec{x})$ which is *preserved under homomorphisms on finite interpretations*. By Rossman's homomorphism preservation theorem [119], we can effectively construct an equivalent positive-existential formula $\varphi''(\vec{x})$. Finally, it is well-known that any positive-existential formula is equivalent to a UCQ. For the construction of $\varphi'(\vec{x})$, we have to take care of two subtle differences between ABoxes and interpretations:

- (i) In an interpretation, two constants might be interpreted as the same domain element, which is not the case for ABoxes due to the unique name assumption.
- (ii) In an interpretation, there might occur elements that are not contained in the interpretation of any relational symbol; such individuals cannot be reflected in an ABox.

For dealing with Point (i), let Σ_c be the set of all constants appearing in φ . It is easy to verify that if φ is a FO-rewriting, then so is $\varphi' = \varphi \vee \theta$ with

$$\theta = \bigvee_{a, b \in \Sigma_c, a \neq b} a = b.$$

In order to show that φ' is preserved under homomorphisms it suffices to consider interpretations over the signature of all constant and relation symbols appearing in φ . A *homomorphism* between two interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ is a mapping $h : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{J}}$ such that

- $h(c^{\mathcal{I}}) = c^{\mathcal{J}}$ for all constant symbols c ;
- $a \in A^{\mathcal{I}}$ implies $h(a) \in A^{\mathcal{J}}$ and $(a, b) \in r^{\mathcal{I}}$ implies $(h(a), h(b)) \in r^{\mathcal{J}}$.

To show that φ' is preserved under homomorphisms on finite interpretations, we assume finite interpretations \mathcal{I} and \mathcal{J} , a tuple $\vec{a} \in (\Delta^{\mathcal{I}})^k$ with $\vec{a} \in \text{ans}(\varphi', \mathcal{I})$, and a homomorphism h from \mathcal{I} to \mathcal{J} . We show $h(\vec{a}) \in \text{ans}(\varphi', \mathcal{J})$.

Assume first that there are distinct constant symbols $c, d \in \Sigma_c$ such that either $c^{\mathcal{I}} = d^{\mathcal{I}}$ or $c^{\mathcal{J}} = d^{\mathcal{J}}$. Note that the former implies the latter as h is an homomorphism from \mathcal{I} to \mathcal{J} . In both cases $\mathcal{J} \models \theta$ and thus $h(\vec{a}) \in \text{ans}(\varphi', \mathcal{J})$. Hence, in what follows assume that both in \mathcal{I} and in \mathcal{J} constants are interpreted as pairwise different domain elements. In particular, we assume that $c^{\mathcal{I}} = c$ and $c^{\mathcal{J}} = c$ for all constant symbols $c \in \Sigma_c$ and that the domains of \mathcal{I} and \mathcal{J} are disjoint otherwise, that is, $\Delta^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = \Sigma_c$. Thus, $\mathcal{J} \not\models \theta$ and it suffices to show that $h(\vec{a}) \in \text{ans}(\varphi, \mathcal{J})$.

By Point (ii) above, there is not necessarily an ABox corresponding to \mathcal{J} . To deal with this, let X be a unary relation symbol not occurring in $\varphi(\vec{x})$, let \mathcal{I}' be the interpretation obtained by extending \mathcal{I} with $X^{\mathcal{I}'} = \Delta^{\mathcal{I}}$, and define \mathcal{J}' analogously. Clearly,

- $\vec{a} \in \text{ans}(\varphi, \mathcal{I})$ iff $\vec{a} \in \text{ans}(\varphi, \mathcal{I}')$;
- $h(\vec{a}) \in \text{ans}(\varphi, \mathcal{J})$ iff $h(\vec{a}) \in \text{ans}(\varphi, \mathcal{J}')$;
- h is still a homomorphism from \mathcal{I}' to \mathcal{J}' .

Note that the construction of \mathcal{I}' ensures the existence of an ABox $\mathcal{A}_{\mathcal{I}'}$ such that $\mathcal{I}' = \mathcal{I}_{(\mathcal{A}_{\mathcal{I}'})}$, and analogously for \mathcal{J}' . As φ is FO-rewriting of $\widehat{\varphi}$ relative to \mathcal{T} , it suffices to show that $h(\vec{a}) \in \text{cert}_{\mathcal{T}}(\widehat{\varphi}, \mathcal{A})$. To see this, we show for an arbitrary model $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$ of \mathcal{T} and $\mathcal{A}_{\mathcal{J}'}$ that $h(a) \in \text{ans}(\varphi, \mathcal{M})$.

Let $\text{Im}_h = \{h(d) \mid d \in \Delta^{\mathcal{I}}\}$ denote the image of h in \mathcal{J} . By construction of $\mathcal{A}_{\mathcal{J}'}$ and the UNA, we have $\text{Im}_h \subseteq \Delta^{\mathcal{M}}$. Next, construct a model $\mathcal{N} = (\Delta^{\mathcal{N}}, \cdot^{\mathcal{N}})$ of $\mathcal{A}_{\mathcal{I}'}$ and \mathcal{T} by starting with the domain

$$\Delta^{\mathcal{N}} = (\Delta^{\mathcal{M}} \setminus \text{Im}_h) \cup \Delta^{\mathcal{I}}.$$

For the definition of $\cdot^{\mathcal{N}}$ we extend the homomorphism h to a mapping $g : \Delta^{\mathcal{N}} \rightarrow \Delta^{\mathcal{M}}$:

$$g(d) = \begin{cases} h(d) & \text{if } d \in \Delta^{\mathcal{I}}; \\ d & \text{otherwise.} \end{cases}$$

Then, define $\cdot^{\mathcal{N}}$ as follows:

- $A^{\mathcal{N}} = \{d \in \Delta^{\mathcal{N}} \mid g(d) \in A^{\mathcal{M}}\}$;
- $r^{\mathcal{N}} = \{(d, e) \in \Delta^{\mathcal{N}} \times \Delta^{\mathcal{N}} \mid (g(d), g(e)) \in r^{\mathcal{M}}\}$;

Clearly, g is a homomorphism from \mathcal{N} to \mathcal{M} . As it is also surjective, it is well-known that we have for every first-order formula $\psi(\vec{x})$ and every valuation ν :

$$(\mathcal{N}, \nu) \models \psi(\vec{x}) \iff (\mathcal{M}, g \circ \nu) \models \psi(\vec{x}).^2 \quad (\dagger)$$

This yields $\mathcal{N} \models \mathcal{T}$ since, by assumption, $\mathcal{M} \models \mathcal{T}$. Next, we show that $\mathcal{N} \models \mathcal{A}_{\mathcal{I}'}$:

² $(g \circ \nu)(x) = g(\nu(x))$ for all x .

- Assume $A(a) \in \mathcal{A}_{\mathcal{I}}$. This implies $a \in A^{\mathcal{I}'}$ and thus $h(a) \in A^{\mathcal{J}'}$. By definition of $\mathcal{A}_{\mathcal{J}}$, we have $A(h(a)) \in \mathcal{A}_{\mathcal{J}}$ and as $\mathcal{M} \models \mathcal{A}_{\mathcal{J}}$, we get $h(a) \in A^{\mathcal{M}}$. By definition of g , we have $h(a) = g(a)$ and hence $a \in A^{\mathcal{N}}$, by Definition of \mathcal{N} .
- For $r(a, b) \in \mathcal{A}_{\mathcal{I}}$, we proceed analogously.

Now, by assumption we have $\mathcal{I}' \not\models \theta$ and thus $\vec{a} \in \text{ans}(\varphi, \mathcal{I}')$. From this, we obtain $\vec{a} \in \text{cert}_{\mathcal{T}}(\widehat{\varphi}, \mathcal{A}_{\mathcal{I}})$, and thus $\vec{a} \in \text{ans}(\widehat{\varphi}, \mathcal{N})$. That is, we have $(\mathcal{N}, \nu) \models \widehat{\varphi}(\vec{x})$ for the valuation that maps x_i to a_i for all $i \in [1, k]$. By Equation (†), we have that $(\mathcal{M}, g \circ \nu) \models \widehat{\varphi}(\vec{x})$. It remains to observe that $g(\nu(x_i)) = g(a_i) = h(a_i)$ for all $i \in [1, k]$ to show that $h(\vec{a}) \in \text{ans}(\widehat{\varphi}, \mathcal{M})$. ■

Now, the proof of Theorem 5.12 is immediate. Fix some φ, \mathcal{T} such that φ is first-order rewritable relative to \mathcal{T} . By Corollary 5.14, we have for *any* first-order rewriting $\varphi_{\mathcal{T}}$, $\text{ipOBDA}(\varphi, \mathcal{T}) = \text{ipOBDA}(\varphi_{\mathcal{T}}, \emptyset)$. By Proposition 5.15, $\varphi_{\mathcal{T}}$ is equivalent to a UCQ-rewriting φ' . Thus, we have $\text{ipOBDA}(\varphi_{\mathcal{T}}, \emptyset) = \text{ipOBDA}(\varphi', \emptyset)$. Finally, Corollary 5.10 implies that $\text{ipOBDA}(\varphi', \emptyset)$ is either in PTIME or #P-hard.

Let us conclude this section with two remarks. First, and disregarding the fact that we study data complexity, let us note that there are no guarantees on the size of the mentioned UCQ-rewriting φ' . Already the first-order rewriting we start with, φ , is only required to be computable. Moreover, Rossman's homomorphism preservation theorem (hidden in the proof of Proposition 5.15) does not yield elementary constructions. However, for restricted TBox and query languages there is hope for better bounds. Second, notice that Theorem 5.12 is very *abstract*, in the sense that it states a strong dichotomy result, but does not tell us *which* CQs are in PTIME relative to *which* TBoxes. This will be subject of the next section.

5.4 The Dichotomy for DL-Lite TBoxes

The goal of this section is to fix a query and a TBox language—conjunctive queries and DL-Lite—and get a better understanding of *which* queries q and TBoxes \mathcal{T} are in PTIME. Since it is well-known that every CQ is first-order rewritable relative to every DL-Lite-TBox [28], the following is an immediate consequence of Theorem 5.12

Corollary 5.16. *Let \mathcal{T} be a DL-Lite-TBox and q a CQ. Then, $\text{ipOBDA}(q, \mathcal{T})$ is either in PTIME or #P-hard.*

Recall that Corollary 5.14 enables us to study the complexity of $\text{ipOBDA}(q, \mathcal{T})$ by looking at the problem $\text{ipOBDA}(\varphi, \emptyset)$ for first-order rewritings φ of q and \mathcal{T} . Thus, in order to understand the complexity of $\text{ipOBDA}(q, \mathcal{T})$, we carry out a careful inspection of FO-rewritings for DL-Lite-TBoxes and the dichotomy for tuple-independent probabilistic databases given in Theorem 5.9. This will result in a concrete formulation of the

dichotomy stated in Corollary 5.16 and provide a transparent characterization of the PTIME cases. For the sake of simplicity, we concentrate on CQs that are connected, Boolean, and do not contain individual names. The general case is left for future work.

For analyzing the complexity of $\text{ipOBDA}(q, \mathcal{T})$ via the FO-rewriting of q relative to \mathcal{T} it is worth noting that, in general, there are different ways to produce an UCQ-rewriting for a given CQ and DL-Lite TBox [28, 26]. However, it is not hard to show that all these rewritings are equivalent. Moreover, when we restrict our attention to *reduced* UCQ-rewritings, that is, UCQs where every CQ is minimal and, if $q \sqsubseteq q'$ for two disjuncts, then $q = q'$, it turns out that there is a *unique* UCQ-rewriting for every CQ q and DL-Lite TBox \mathcal{T} (up to variable renaming). We are going to provide a characterization of this unique rewriting which will be useful later, but before we need some notation. For what follows, fix a TBox \mathcal{T} . For two CQs q, q' , we say that q \mathcal{T} -*implies* q' and write $q \sqsubseteq_{\mathcal{T}} q'$ when $\text{cert}_{\mathcal{T}}(q, \mathcal{A}) \subseteq \text{cert}_{\mathcal{T}}(q', \mathcal{A})$ for all ABoxes \mathcal{A} . Note that for Boolean CQs $q \sqsubseteq_{\mathcal{T}} q'$ if, and only if, $\mathcal{T}, \mathcal{A}_q \models q'$. We say that q and q' are \mathcal{T} -*equivalent* and write $q \equiv_{\mathcal{T}} q'$ if $q \sqsubseteq_{\mathcal{T}} q'$ and $q' \sqsubseteq_{\mathcal{T}} q$. Finally, q is \mathcal{T} -*minimal* if there is no $q' \subsetneq q$ such that $q \equiv_{\mathcal{T}} q'$.

Example 5.17. Fix the TBox $\mathcal{T} = \{A \sqsubseteq \exists r, \exists r^- \sqsubseteq B, A' \sqsubseteq A, \exists r \sqsubseteq A'\}$ and consider the queries:

$$\begin{aligned} q_1 &= \exists xy A(x), r(x, y), B(y) & q_2 &= \exists x A(x) \\ q_3 &= \exists x A'(x) & q_4 &= \exists x A(x), B(x). \end{aligned}$$

It should be clear that $q_4 \sqsubseteq_{\mathcal{T}} q_2$ and $q_1 \sqsubseteq_{\mathcal{T}} q_2$ as q_1, q_4 contain additional atoms compared to q_2 . Further, $q_2 \sqsubseteq_{\mathcal{T}} q_1$ as $\mathcal{T} \models A \sqsubseteq \exists r.B$,³ and thus $q_1 \equiv_{\mathcal{T}} q_2$. However, $q_2 \not\sqsubseteq_{\mathcal{T}} q_4$ as for $\mathcal{A} = \{A(a)\}$ we have $\mathcal{T}, \mathcal{A} \models q_2$ but $\mathcal{T}, \mathcal{A} \not\models q_4$. Further note that $q_2 \equiv_{\mathcal{T}} q_3$ as $\mathcal{T} \models A \equiv A'$. In fact, $\equiv_{\mathcal{T}}$ is an equivalence relation and thus $q_1 \equiv_{\mathcal{T}} q_2 \equiv_{\mathcal{T}} q_3$. Finally note that minimality is not a total order, in particular, both q_2 and q_3 are \mathcal{T} -minimal.

To have more control over the effect of the TBox, we will generally work with CQs q and TBoxes \mathcal{T} such that q is \mathcal{T} -minimal. This is without loss of generality because for every TBox \mathcal{T} and any two \mathcal{T} -equivalent CQs q, q' , we have that the answer probabilities relative to \mathcal{T} are identical for q and q' and indeed $\text{ipOBDA}(q, \mathcal{T}) = \text{ipOBDA}(q', \mathcal{T})$. Moreover, for many TBox languages including DL-Lite, we can effectively find a CQ q' that is \mathcal{T} -minimal and such that $q \equiv_{\mathcal{T}} q'$ [16].

The next lemma provides the promised characterization of the unique reduced UCQ-rewriting, which is denoted with $q_{\mathcal{T}}$ in what follows. Note that the proof of this theorem does not depend on the TBox language; it is not hard to verify that it goes through for full first-order logic (without equality). Let us also remark that this lemma is not constructive in the sense that it does not immediately give us a FO-rewriting. However, it gives us a condition that we can work with later.

³ $\exists r.B$ is not a DL-Lite concept, but it should be clear what is meant.

Lemma 5.18. *For each conjunctive query q and DL-Lite-TBox \mathcal{T} , there is a unique reduced UCQ-rewriting $q_{\mathcal{T}}$. In particular, $q_{\mathcal{T}}$ consists of all q' with $q' \sqsubseteq_{\mathcal{T}} q$ such that*

(*) *there is no CQ $q'' \not\equiv q'$ with $q' \sqsubseteq q''$ and $q'' \sqsubseteq_{\mathcal{T}} q$.*

Proof. “ \supseteq ”: Let q' be a CQ with $q' \sqsubseteq_{\mathcal{T}} q$ such that there is no $q'' \not\equiv q'$ with $q' \sqsubseteq q''$ and $q'' \sqsubseteq_{\mathcal{T}} q$. Note first that this implies minimality of q' , that is, there is no strict sub-query of q' that is equivalent to q' . Now, take any reduced UCQ-rewriting φ . As $q' \sqsubseteq_{\mathcal{T}} q$, we have that $\mathcal{T}, \mathcal{A}_{q'} \models q$ and, since φ is FO-rewriting, $\mathcal{I}_{\mathcal{A}_{q'}} \models \varphi$. Hence, there is some disjunct p of φ such that there is a homomorphism from p to q' , thus $q' \sqsubseteq p$. As p is a disjunct of φ , we must have that $p \sqsubseteq_{\mathcal{T}} q$. By our initial assumption, we get $q' \equiv p$. As both q' and p are minimal, we get that in fact $p = q'$, thus q' is a disjunct of φ .

“ \subseteq ”: Let q' be a disjunct of some reduced UCQ-rewriting φ . Clearly, we have $q' \sqsubseteq_{\mathcal{T}} q$. So assume that there is some $q'' \not\equiv q'$ with $q' \sqsubseteq q''$ and $q'' \sqsubseteq_{\mathcal{T}} q$. By the latter, we have $\mathcal{T}, \mathcal{A}_{q''} \models q$. As φ is a FO-rewriting, this yields $\mathcal{I}_{\mathcal{A}_{q''}} \models \varphi$. Hence, there is some disjunct p of φ such that there is a homomorphism from p to q'' , thus $q'' \sqsubseteq p$. As $q' \sqsubseteq q''$, we also have $q' \sqsubseteq p$. Since φ is reduced, we get $q' = p$. Because of $q'' \sqsubseteq p$, this implies $q'' \sqsubseteq q'$, hence $q' \equiv q''$, contradiction. ■

As already mentioned, our approach to study the complexity of $\text{ipOBDA}(q, \mathcal{T})$ is to look at $\text{ipOBDA}(q_{\mathcal{T}}, \emptyset)$ in the light of the dichotomy Theorem 5.9 for UCQs. However, also Theorem 5.9 is abstract, so we need to substantiate it. As it is well-known that each disjunct of $q_{\mathcal{T}}$ is connected when q is connected, it suffices to consider the dichotomy for the subcase where every disjunct is a connected CQ. We refer to such UCQs as *disjunctive sentences*, in order to be consistent with [37].

5.4.1 Dichotomy for disjunctive sentences

We first need some additional notation. As mentioned, a disjunctive sentence is a formula

$$q = \exists x_1 \dots x_n (q_1 \vee \dots \vee q_k).$$

where each q_ℓ is a conjunction of atoms $A(x_i)$ or $r(x_i, x_j)$, that is, we disallow constants. We will usually omit the quantifiers and implicitly assume that every variable is existentially quantified. We call a disjunctive sentence $q = q_1 \vee \dots \vee q_k$ *reduced* if:

- each q_i is minimal and
- $q_i \sqsubseteq q_j$ implies $i = j$ for all i, j .

As we can effectively reduce a disjunctive sentence [29] and the resulting sentence has the same complexity, we will restrict our attention to reduced sentences.

A *root variable* of a disjunctive sentence q is a variable that occurs in all atoms of q . In what follows, we assume that, if a disjunctive sentence can be equivalently rewritten

(by renaming variables) into one having a root variable, then it is rewritten. For example, the query $A(x) \vee B(y)$ is (equivalently) rewritten as $A(x) \vee B(x)$, and thus has a root variable. A root variable x of q is a *weak separator variable* of q if additionally for every binary relation name r that occurs in the query, there is a number $i_r \in \{1, 2\}$ such that every atom $r(x_1, x_2)$ satisfies $x_{i_r} = x$. A weak separator variable is a *separator variable* for q when no atom of the form $r(x, x)$ appears in q . As example, observe that $r(x, y), r(y, z)$ does have a root variable y , but no weak separator variable since y appears at position 2 in $r(x, y)$ and at position 1 in $r(y, z)$. Moreover, the $r(x, y), r(y, y)$ has the weak separator variable y (choose $i_r = 2$), but no separator variable because of $r(y, y)$. An example for a query with separator variable is $A(x), r(x, y), s(z, x)$.

Recall that $R(x, y)$ is either $r(x, y)$ or $r(y, x)$ for some role name r and that $p_{\mathcal{A}}^d(q)$ denotes the probability of q in \mathcal{A} viewed as a tuple-independent probabilistic database. The dichotomy for disjunctive sentences is as follows.

Theorem 5.19 (Concrete Dichotomy for Disjunctive Sentences). *Let q be a reduced disjunctive sentence. Then computing $p_{\mathcal{A}}^d(q)$ is in PTIME if:*

- (i) *each disjunct of q is of the form $R_1(x, y), \dots, R_k(x, y)$; or*
- (ii) *q has a weak separator variable.*

Otherwise, it is #P-hard.

Consequences of this theorem are the facts that $A(x), r(x, y) \vee B(x), r(z, x)$ is #P-hard, but $A(x), r(x, y) \vee B(x), r(x, x)$ is in PTIME.

The proof of this theorem is adapted from [37]. We include it here in order to be self-contained and because Theorem 5.19 is not explicit in [37]. We start with the PTIME case for which we need further notation. A disjunctive sentence $q = q_1 \vee \dots \vee q_k$ is *symbol-connected* if the graph (V, E) with

- $V = \{q_1, \dots, q_k\}$, and
- $\{q_i, q_j\} \in E$ iff there is a relation name that occurs both in q_i and q_j

is connected. If a disjunctive sentence is not symbol-connected, we can read off its *connected components* q_1, \dots, q_ℓ from this graph (V, E) .

We are going to give a polynomial time algorithm that computes $p_{\mathcal{A}}^d(q)$. For the sake of simplicity, let us start with the case that there is a separator variable x^* , that is, the subcase of condition (ii) where no atoms of the form $r(x^*, x^*)$ appear. In order to compute $p_{\mathcal{A}}^d(q)$, we will stepwise apply one of the rules depicted in Figure 5.1, where $[n]$ denotes the set $\{1, \dots, n\}$, and $\binom{[n]}{k}$ denotes the set of k -elementary subsets of $[n]$. The rules are taken from [37] and their soundness is easy to verify. Let us only look at the most interesting case, **Independent Project**. As x is a separator variable, any two

Equivalence Transformation We can transform q to any equivalent query.

Independent Join If the query q is a conjunction of symbol-disconnected queries q_1, \dots, q_n , then compute $p_{\mathcal{A}}^d(q) = p_{\mathcal{A}}^d(q_1) \cdot \dots \cdot p_{\mathcal{A}}^d(q_n)$.

Independent Union If the query q is a disjunction of symbol-disconnected queries q_1, \dots, q_n , then compute $p_{\mathcal{A}}^d(q) = 1 - (1 - p_{\mathcal{A}}^d(q_1)) \cdot \dots \cdot (1 - p_{\mathcal{A}}^d(q_n))$.

Independent Project If the query has a separator variable x , then compute $p_{\mathcal{A}}^d(q) = 1 - \prod_{a \in \text{Ind}(\mathcal{A})} (1 - p_{\mathcal{A}}^d(q[a/x]))$.

Inclusion-exclusion principle If q is a disjunction of queries q_1, \dots, q_n , then its probability can be computed by:

$$p_{\mathcal{A}}^d(q_1 \vee \dots \vee q_n) = \sum_{k=1}^n (-1)^{k-1} \sum_{I \in \binom{[n]}{k}} p_{\mathcal{A}}^d(\bigwedge_{i \in I} q_i).$$

There is a dual rule which is obtained by exchanging \vee and \wedge in the above rule (justified by the dual inclusion-exclusion principle).

Figure 5.1: The rules for computing query probabilities.

atoms $\mathbf{at} \in q[a/x]$ and $\mathbf{at}' \in q[b/x]$ cannot be unified (by replacing free variables with constants). Thus, the queries $q[a/x]$ and $q[b/x]$ are syntactically independent for $a \neq b$ which yields soundness of the rule. The algorithm proceeds as follows.

1. Let q_1, \dots, q_k be the symbol-connected components of q . We apply the rule **Independent Union**:

$$p_{\mathcal{A}}^d(q) = 1 - (1 - p_{\mathcal{A}}^d(q_1)) \cdot \dots \cdot (1 - p_{\mathcal{A}}^d(q_k)).$$

2. Each q_i has the separator variable x^* . We apply **Independent Project**:

$$p_{\mathcal{A}}^d(q_i) = 1 - \prod_{a \in \text{Ind}(\mathcal{A})} (1 - p_{\mathcal{A}}^d(q_i[a/x^*])).$$

3. Each disjunct of $q_i[a/x^*]$ can be written in the form

$$A_1(a), \dots, A_m(a), \bar{p}_1(z_1), \dots, \bar{p}_n(z_n) \quad (*)$$

where the z_i are pairwise different variables, and each \bar{p}_i contains only atoms of the form $S(a, z_i)$. Note that, semantically, the commas in $(*)$ are in fact conjunctions

“ \wedge ”. Let $q'_i[a]$ be obtained from $q_i[a/x^*]$ by distributing the outermost \vee of $q_i[a/x^*]$ over these conjunctions (but not inside the $\bar{p}_i(z_i)$). This is an instance of rule **Equivalence Transformation**.

4. The query $q'_i[a]$ is a conjunction of, say N , disjunctive sentences p_1, \dots, p_N each of whose disjuncts is of the form $A(a)$ or $S_1(a, z), \dots, S_\ell(a, z)$. The probability $p_{\mathcal{A}}^d(q'_i[a])$ is computed using the (dual) **Inclusion-exclusion principle**:

$$p_{\mathcal{A}}^d(p_1 \wedge \dots \wedge p_N) = \sum_{k=1}^N (-1)^{k-1} \sum_{I \in \binom{[N]}{k}} p_{\mathcal{A}}^d(\bigvee_{i \in I} p_i).$$

5. The structure of $q_I := \bigvee_{i \in I} p_i$ for some non-empty $I \subseteq [N]$ is as follows:

$$A_1(a) \vee \dots \vee A_k(a) \vee q'_I,$$

where q'_I is a disjunction of connected CQs of the form $S_1(a, z), \dots, S_\ell(a, z)$. Using **Independent Union**, we compute:

$$p_{\mathcal{A}}^d(q_I) = 1 - (1 - p_{\mathcal{A}}^d(q'_I)) \cdot \prod_{i=1}^k (1 - p_{\mathcal{A}}^d(A_i(a))).$$

6. Observe that $p_{\mathcal{A}}^d(A_i(a))$ can be read off from \mathcal{A} . Hence, it remains to compute $p_{\mathcal{A}}^d(q'_I)$. Since every atom has precisely one free variable, q'_I can be viewed as a query with a separator z . By rule **Independent Project**, we obtain:

$$p(q'_I) = 1 - \prod_{b \in \text{Ind}(\mathcal{A})} (1 - p(q'_I[b/z])).$$

7. The query $q'_I[b/z]$ is the disjunction of, say M , conjunctive queries d_1, \dots, d_M , each of the form $S_1(a, b), \dots, S_\ell(a, b)$. By the **Inclusion-exclusion principle** we can compute:

$$p_{\mathcal{A}}^d(d_1 \vee \dots \vee d_M) = \sum_{k=1}^M (-1)^{k-1} \sum_{I \in \binom{[M]}{k}} p_{\mathcal{A}}^d(\bigwedge_{i \in I} d_i).$$

8. For any non-empty $I \subseteq [M]$, the query $\bigwedge_{i \in I} d_i$ can be viewed as a collection of atoms of the form $S(a, b)$. As all atoms are ground, we can apply **Independent Join** to compute:

$$p_{\mathcal{A}}^d(\bigwedge_{i \in I} d_i) = \prod_{S(a, b) \in \bigcup_{i \in I} d_i} p_{\mathcal{A}}^d(S(a, b)).$$

Let us analyze the running time of this algorithm. Note that only in Steps 2 and 6, the input is taken into account, by creating instances of the queries for every $a \in \text{Ind}(\mathcal{A})$; this is linear. In all other steps, we apply rules only depending on the (fixed) query q . Thus, the algorithm runs in polynomial time.

We now show how to amend the algorithm in order to cover the more general conditions (i) and (ii). We first deal with condition (i). If q is of the form in condition (1) and additionally has a separator variable x^* , we are done as we can run the above steps. Otherwise, q contains atoms $r(x^*, y)$ and $r(y, x^*)$. For handling those, we need the following notion. We say that a query \hat{q} is ranked, if the set

$$\{x < y \mid r(x, y) \text{ is atom in } \hat{q}\}$$

can be extended to a total order. In particular, q is not ranked as the set necessarily contains $x^* < y$ and $y < x^*$ which cannot be jointly satisfied by any total order. The following proposition tells us that we can ‘rank’ each query.

Proposition 5.20 (Proposition 4.2 of [37]). *Every disjunctive query q is equivalent (up to polynomial time many-one reductions) to a ranked disjunctive query \bar{q} over an extended vocabulary. More specifically, the problem “given \mathcal{A} , compute $p_{\mathcal{A}}^d(q)$ ” can be reduced in polynomial time to “given $\bar{\mathcal{A}}$, compute $p_{\bar{\mathcal{A}}}(\bar{q})$ ”, and vice versa.*

Proof. We assume that q has variables $X = \{x_1, \dots, x_n\}$. Fix a total order $x_1 < \dots < x_n$ on X and denote $X_p = \{x_1, \dots, x_p\}$ for $p \leq n$. A *variable ranking* ρ is a surjective map $\rho : X \rightarrow X_p$ for some $p \leq n$. Obtain the query q_ρ from q by replacing every atom at with at_ρ where:

- $A(x_i)_\rho = A(\rho(x_i))$;
- $r(x_i, x_j)_\rho = \begin{cases} r^=(\rho(x_i)) & \text{if } \rho(x_i) = \rho(x_j); \\ r^<(\rho(x_i), \rho(x_j)) & \text{if } \rho(x_i) < \rho(x_j); \\ r^>(\rho(x_j), \rho(x_i)) & \text{if } \rho(x_j) < \rho(x_i). \end{cases}$

The query \bar{q} is defined as $\bigvee_\rho q_\rho$. Observe that each disjunct of q_ρ is connected when every disjunct of q is connected. Assume some total order \prec on the domain and obtain $\bar{\mathcal{A}}$ as follows:

- carry over all assertions of the form $A(a)$ from \mathcal{A} ;
- For each assertion $r(a, a) \in \mathcal{A}$, add $r^=(a)$ to $\bar{\mathcal{A}}$;
- For each assertion $r(a, b) \in \mathcal{A}$ with $a \prec b$, add $r^<(a, b)$ to $\bar{\mathcal{A}}$;
- For each assertion $r(a, b) \in \mathcal{A}$ with $a \succ b$, add $r^>(b, a)$ to $\bar{\mathcal{A}}$;

- transfer all associated events in the obvious way.

It should be clear that $p_{\mathcal{A}}^d(q) = p_{\mathcal{A}}^d(\bar{q})$ and that we can reconstruct \mathcal{A} , q from $\bar{\mathcal{A}}$, \bar{q} in polynomial time. \blacksquare

Example 5.21. Consider the unranked query $q = r(x_1, x_2), r(x_2, x_1)$. There are three variable rankings ρ_1, ρ_2, ρ_3 given by $\rho_1(x_1, x_2) = (x_1, x_2)$, $\rho_2(x_1, x_2) = (x_2, x_1)$, and $\rho_3(x_1, x_2) = (x_1, x_1)$. Thus, the ranked query \bar{q} is the disjunction of the following three queries:

$$\begin{aligned} q_{\rho_1} &= r^{\prec}(x_1, x_2), r^{\succ}(x_1, x_2); \\ q_{\rho_2} &= r^{\succ}(x_1, x_2), r^{\prec}(x_1, x_2); \\ q_{\rho_3} &= r^=(x_1), r^=(x_1). \end{aligned}$$

Moreover, after reducing it, we get $\bar{q} = r^=(x_1) \vee r^{\prec}(x_1, x_2), r^{\succ}(x_1, x_2)$.

Turning back to our input query q , we generalize Example 5.21 and obtain that every disjunct of \bar{q} is of the form:

$$R_1^=(x^*), \dots, R_k^=(x^*) \vee R_1^{\prec}(x^*, y), \dots, R_k^{\prec}(x^*, y) \vee R_1^{\succ}(x^*, y), \dots, R_k^{\succ}(x^*, y),$$

where $R_i^= = s_i^=$ if $R_i \in \{s_i, s_i^-\}$, and R_i^{\prec} is s_i^{\prec} if $R_i = s_i$ and s_i^{\succ} otherwise, and analogously for R_i^{\succ} . Because x^* is in the first position of every appearing atom, x^* is a separator in \bar{q} . Moreover, note that \bar{q} does not contain atoms of the form $r(x^*, x^*)$. Hence, $p_{\bar{\mathcal{A}}}^d(\bar{q})$ can be computed using the basic algorithm.

Assume finally that q is not of the form under point (i) of the theorem but has only a weak separator variable. That is, q has a unique root variable x^* but there are atoms of the form $r(x^*, x^*)$ in q . Consider again its ranking \bar{q} . By construction, in the ranking there are no atoms of the form $r(x^*, x^*)$. Assume now that there is a relation name s^{\prec} such that there are disjuncts q_1, q_2 in \bar{q} such that $s^{\prec}(x^*, y_1) \in q_1$ with $x^* \neq y_1$ and $s^{\prec}(y_2, x^*) \in q_2$ with $x^* \neq y_2$. By the construction in Proposition 5.20, there are disjuncts q'_1, q'_2 in q such that $s(x^*, y_1) \in q_1$ with $x^* \neq y_1$ and $s(y_2, x^*) \in q_2$ with $x^* \neq y_2$. This is a contradiction, since then either q is of the form under point (i) or q does not have a weak separator variable. This finishes the proof of the PTIME-part of the theorem.

Before giving details of the #P-hardness part, let us start with some intuition. The ‘prototypical’ #P-hard query is:

$$q_{\perp} = \exists xy A(x), r(x, y), B(y).$$

Clearly, q_{\perp} has no root variable and thus does not satisfy conditions (i) or (ii). The proof is via reduction from #MonBiDNF which is #P-hard by Theorem 5.5. Given a formula $\varphi = \bigvee_i (x_i \wedge y_i)$ over bipartite variable sets X and Y , define an ipABox \mathcal{A} as follows. \mathcal{A} consists of assertions $A(x)$ and $B(y)$ for all variables $x \in X$ and $y \in Y$. All

these assertions are annotated with probability $1/2$. Moreover, for each clause $x \wedge y$ in φ , \mathcal{A} contains the assertion $r(x, y)$ with probability 1. Then, we have that the probability $p_{\mathcal{A}}^d(q_{\perp})$ is precisely $\#\varphi/2^n$, where n is the number of variables in φ . This prototypical query q_{\perp} justifies the necessity of the existence of a root variable (note that a root variable is required in both conditions for PTIME). The fact that a root variable in condition (ii) is insufficient is motivated by the fact that one can similarly show $\#P$ -hardness of the query $\exists xyz r(x, y), r(y, z)$ where y is root but not a separator variable.

For the full proof of hardness, it is important to know how the original dichotomy from [37] is proved. Basically, (a superset of) the rules in Figure 5.1 are applied to a ranked query q just as it was done above. When no rule is applicable anymore, the algorithm returns ‘fail’ meaning that q is $\#P$ -hard. We next introduce the condition of when no rule is applicable anymore. We say that a reduced disjunctive sentence is *immediately unsafe* if it is ranked, symbol-connected, contains at least one variable, but does not have a separator.

Theorem 5.22 (Theorem 4.5 of [37]). *If a reduced disjunctive query q is immediately unsafe, then computing $p_{\mathcal{A}}^d(q)$ on input \mathcal{A} is $\#P$ -hard.*

This theorem will be the basis for the $\#P$ -hardness part. So assume that q does not satisfy condition (i) or condition (ii), that is,

- (a) it has no root variable, or
- (b) it has a unique root variable x^* , but this variable is not a weak separator.

We cannot apply Theorem 5.22 directly, as q is not necessarily ranked. By Proposition 5.20, we can consider its ranking \bar{q} . Note that Example 5.21 shows that \bar{q} might not be reduced even if q is reduced, which again prevents application of Theorem 5.22. However, in the subsequent lemma, we show that certain properties of \bar{q} are invariant under reduction.

Lemma 5.23. *If q is a reduced disjunctive sentence and p is some disjunct in q , then p_{τ} is contained in the reduction of \bar{q} for every injective variable ranking τ .*

Proof. We show the statement for the variable ranking $\tau = \text{id}$ which maps every variable to itself, and note that the proof extends to all injective variable rankings.

Suppose to the contrary that there is some disjunct p' in q and a variable ranking ρ such that $p_{\text{id}} \sqsubseteq p'_{\rho}$ but $p_{\text{id}} \neq p'_{\rho}$. Thus, there is a homomorphism h from p'_{ρ} to p_{id} . We claim that $h \circ \rho$ is a homomorphism from p' to p , contradicting q being reduced.

- If $A(x) \in p'$, then $A(\rho(x)) \in p'_{\rho}$ and since h is homomorphism, we have also $A(h(\rho(x))) \in p_{\rho}$. Thus, $A(h(\rho(x))) \in p$.
- For $r(x, y) \in p'$, we distinguish cases:

- If $\rho(x) < \rho(y)$, then $r^{\prec}(\rho(x), \rho(y)) \in p'_\rho$. Because h is a homomorphism, we have $r^{\prec}(h(\rho(x)), h(\rho(y))) \in p_{\text{id}}$. By definition of p_{id} , there is some atom $r(h(\rho(x)), h(\rho(y))) \in p$.
- If $\rho(x) > \rho(y)$, then $r^{\succ}(\rho(y), \rho(x)) \in p'_\rho$. Because h is a homomorphism, we have $r^{\succ}(h(\rho(y)), h(\rho(x))) \in p_{\text{id}}$. By definition of p_{id} , there is some atom $r(h(\rho(x)), h(\rho(y))) \in p$.
- If $\rho(x) = \rho(y)$, then $r^=(\rho(x)) \in p'_\rho$. Because h is a homomorphism, we have $r^=(h(\rho(x))) \in p_{\text{id}}$, contradicting the definition of p_{id} .

■

We are now coming back to our input query q satisfying (a) or (b) and let \bar{q} be the reduced ranking of q . In case of (a), there is a disjunct p in q that does not have a root variable. By Lemma 5.23, the disjunct p_{id} is also in \bar{q} . Thus, \bar{q} does not have a root variable and hence it does not have a separator.

In case of (b), there are disjuncts q_1, q_2 of q , variables y_1, y_2 , and a relation name r such that

- x^* is unique root variable in q_1 , $r(x^*, y_1) \in q_1$, and $y_1 \neq x^*$; and
- x^* is unique root variable in q_2 , $r(y_2, x^*) \in q_2$, and $y_2 \neq x^*$.

Let ρ, ρ' be injective variable rankings with $\rho(x^*) < \rho(y_1)$ and $\rho'(y_2) < \rho'(x^*)$. By Lemma 5.23, \bar{q} contains disjuncts $q_\rho, q_{\rho'}$ such that:

- x^* is unique root variable in q_ρ , $r^{\prec}(x^*, y_1) \in q_\rho$, and $y_1 \neq x^*$; and
- x^* is unique root variable in $q_{\rho'}$, $r^{\prec}(y_2, x^*) \in q_{\rho'}$, and $y_2 \neq x^*$.

Hence also in case (b), \bar{q} does not have a separator.

If \bar{q} is symbol-connected, \bar{q} is #P-hard via Theorem 5.22. By Proposition 5.20, q has the same complexity as \bar{q} and is thus also #P-hard. If not, let q_1, \dots, q_k be its connected-components. The algorithm from [37] applies rule **Independent Union** to get

$$p_{\mathcal{A}}^d(\bar{q}) = 1 - (1 - p_{\mathcal{A}}^d(q_1)) \cdot \dots \cdot (1 - p_{\mathcal{A}}^d(q_k)).$$

As \bar{q} does not have a separator, one of the q_i does not have a separator. By Theorem 5.22, q_i is #P-hard. By completeness of the algorithm, also \bar{q} is #P-hard. By Proposition 5.20, also q is #P-hard.

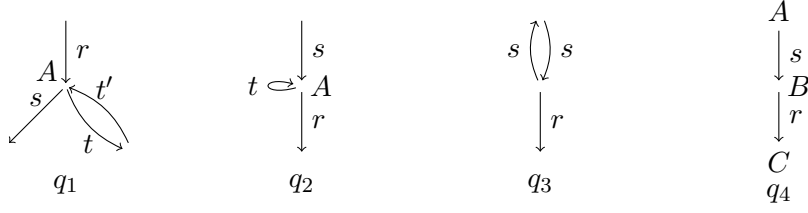


Figure 5.2: Example queries

5.4.2 Dichotomy for DL-Lite

We can now continue to show our characterization of the tractable pairs (q, \mathcal{T}) for DL-Lite-TBoxes \mathcal{T} and CQs q . Inspired by Theorem 5.19, we introduce a class of queries that will play a crucial role in our analysis.

Definition 5.24 (Simple Tree Queries). A CQ q is a simple tree if it has a root variable, that is, a variable that occurs in all atoms in q .

As examples, consider the CQs in Figure 5.2. Queries q_1, q_2, q_3 are all simple tree queries whereas q_4 is not: it contains atoms $A(x), B(y), C(z)$ and thus no root variable. Recall that, by Theorem 5.19, not being a simple tree query is a sufficient condition for being intractable in the framework of probabilistic databases. We will first prove that this lifts to our framework.

Theorem 5.25. Let q be a CQ and \mathcal{T} a DL-Lite TBox such that q is \mathcal{T} -minimal. If q is not a simple tree query, then $\text{ipOBDA}(q, \mathcal{T})$ is $\#P$ -hard.

Proof. We start by showing that the UCQ-rewriting $q_{\mathcal{T}}$ of q relative to \mathcal{T} contains a disjunct which is not a simple tree query. By Lemma 5.18, there is a disjunct p such that $q \sqsubseteq p$ and $p \sqsubseteq_{\mathcal{T}} q$. Thus, there is a homomorphism h from p to q .

- If h is not atom-surjective, then there is a subquery $q' \subsetneq q$ such that $q' \sqsubseteq p$. As $p \sqsubseteq_{\mathcal{T}} q$, we have $q' \sqsubseteq_{\mathcal{T}} q$, contradicting \mathcal{T} -minimality of q .
- Assume now that p is a simple tree query, that is, there is a variable x appearing in all atoms of p . Obviously, $h(x)$ appears in all atoms of q as h is atom-surjective. Thus, q is a simple tree query, contradiction.

Having established this, note that every disjunct of $q_{\mathcal{T}}$ is a connected query if q itself is connected. Thus, we can assume that $q_{\mathcal{T}}$ is written as a disjunctive sentence, which still contains p as a disjunct. However, as p is not a simple tree query, $q_{\mathcal{T}}$ cannot have a root variable. By Theorem 5.19, the problem of computing $p_{\mathcal{A}}^d(q_{\mathcal{T}})$ is $\#P$ -hard. By Observation 5.8, this is the same problem as computing $p_{\mathcal{A}}(q_{\mathcal{T}})$ which is in turn equivalent to $\text{ipOBDA}(q, \mathcal{T})$ by Corollary 5.14. ■

Hence, in order to obtain a dichotomy, it remains to analyze simple tree queries. The central notion for this purpose is introduced next. We say that a role R (a role name r or its inverse r^-) is \mathcal{T} -generated in a simple tree query q if one of the following holds:

- (i) $R(x, y) \sqsubseteq_{\mathcal{T}} q$;
- (ii) there is an atom $R(x, y) \in q$ with $x \neq y$ and x root variable;
- (iii) there is an atom $A(x) \in q$ and $\mathcal{T} \models \exists R \sqsubseteq A$ and x root variable;
- (iv) there is an atom $S(x, y) \in q$ such that $y \neq x$ occurs only in this atom, $\mathcal{T} \models \exists R \sqsubseteq \exists S$, and x root variable.

As examples, consider the queries in Figure 5.2. The role r is \emptyset -generated in q_2 and q_3 by item (ii) above. For the TBox $\mathcal{T} = \{\exists s \sqsubseteq \exists r\}$, note that both q_2 and q_3 \mathcal{T} -generate s because of item (iv) above. For (i) consider the TBox $\mathcal{T} = \{\exists r^- \sqsubseteq \exists s, \exists s^- \sqsubseteq \exists t\}$ and the query $\exists xy t(x, y)$. We then have $r(x, y) \sqsubseteq_{\mathcal{T}} q$, thus r is \mathcal{T} -generated.

The concrete version of the DL-Lite-dichotomy is now as follows.

Theorem 5.26 (Concrete Dichotomy for DL-Lite). *Let \mathcal{T} be a DL-Lite TBox and q a \mathcal{T} -minimal CQ. Then, $\text{ipOBDA}(q, \mathcal{T})$ is in PTIME if it is a simple tree query and one of the following is satisfied:*

- (1) q is of the form $S_1(x, y), \dots, S_k(x, y)$ for roles S_1, \dots, S_k ;
- (2) for all role names r : if both r and r^- are \mathcal{T} -generated in q , then $r(x, y) \sqsubseteq_{\mathcal{T}} q$.

Otherwise, $\text{ipOBDA}(q, \mathcal{T})$ is #P-hard.

Example 5.27. As exemplary consequences of Theorem 5.26, consider again the queries q_1 , q_2 , and q_3 in Figure 5.2 and let \mathcal{T}_\emptyset be the empty TBox. All CQs are \mathcal{T}_\emptyset -minimal, and q_1 and q_2 are in PTIME. On the other hand, q_3 is #P-hard as it is not of the form $S_1(x, y), \dots, S_k(x, y)$ and both s and s^- are \mathcal{T}_\emptyset -generated but $s(x, y) \not\sqsubseteq_{\mathcal{T}_\emptyset} q$. Now consider the TBox $\mathcal{T} = \{\exists s \sqsubseteq \exists r\}$. Then q_1 is \mathcal{T} -minimal and still in PTIME; q_2 is \mathcal{T} -minimal, and is now #P-hard because both s and s^- is \mathcal{T} -generated. The CQ q_3 can be made \mathcal{T} -minimal by dropping the r -atom, and is in PTIME relative to \mathcal{T} . Thus, the TBox has influence on the complexity of a query q in all possible ways: q can become harder in presence of \mathcal{T} , or easier, or stay the same.

Note that condition (1) of Theorem 5.26 corresponds to condition (1) of Theorem 5.19. To see that condition (2) above reflects to condition (2) in Theorem 5.19, we prove the following lemma justifying the notion of ‘being \mathcal{T} -generated’. Its proof is based on a careful analysis of FO-rewritings.

Lemma 5.28. *Let \mathcal{T} be a DL-Lite TBox, q be a \mathcal{T} -minimal simple tree query, and $q_{\mathcal{T}}$ their reduced UCQ-rewriting. For every role R , the following are equivalent:*

- R is \mathcal{T} -generated;
- there is a disjunct q' of $q_{\mathcal{T}}$ that contains an atom $R(x, y)$ where x is a root variable of q' and $x \neq y$.

For proving Lemma 5.28, it is helpful to characterize the notion of \mathcal{T} -implication using *canonical models* as defined in [94]. To construct the canonical model $\mathcal{I}_{\mathcal{T}, \mathcal{A}}$ of an ABox \mathcal{A} and a DL-Lite TBox \mathcal{T} , we start with \mathcal{A} viewed as an interpretation and then exhaustively apply the CIs from \mathcal{T} as rules, introducing fresh elements for existential quantifiers. Formally, the domain of $\mathcal{I}_{\mathcal{T}, \mathcal{A}}$ consists of all paths of the form $aR_1 \cdots R_n$, $n \geq 0$, such that the following conditions hold:

(agen) $\mathcal{T}, \mathcal{A} \models \exists R_1(a)$ but $R_1(a, b) \notin \mathcal{A}$ for all $b \in \text{Ind}(\mathcal{A})$ (written $a \rightsquigarrow c_{R_1}$);

(rgen) for $i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$ (written $c_{R_i} \rightsquigarrow c_{R_{i+1}}$).

We denote by $\text{tail}(\sigma)$ the last element in a path σ . Now, $\mathcal{I}_{\mathcal{T}, \mathcal{A}}$ is defined as follows:

$$\begin{aligned} \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} &= \{a \cdot c_{R_1} \cdots c_{R_n} \mid a \in \text{Ind}(\mathcal{A}), n \geq 0, a \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n}\}, \\ a^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} &= a, \text{ for all } a \in \text{Ind}(\mathcal{A}), \\ A^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} &= \{a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \{\sigma \cdot R \in \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} \mid \mathcal{T} \models \exists R^- \sqsubseteq A\}, \\ P^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} &= \{(a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A}\} \cup \\ &\quad \{(\sigma, \sigma \cdot c_P) \in \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} \times \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} \mid \text{tail}(\sigma) \rightsquigarrow c_P\} \cup \\ &\quad \{(\sigma \cdot c_{P-}, \sigma) \in \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} \times \Delta^{\mathcal{I}_{\mathcal{T}, \mathcal{A}}} \mid \text{tail}(\sigma) \rightsquigarrow c_{P-}\}, \end{aligned}$$

where ‘ \cdot ’ denotes concatenation. The following is the central property of canonical models.

Theorem 5.29 ([94]). *For every consistent DL-Lite KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and every CQ q , we have $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{I}_{\mathcal{T}, \mathcal{A}} \models q$.*

We will sometimes also use canonical models $\mathcal{I}_{\mathcal{T}, q}$ for a CQ q and a TBox \mathcal{T} , defined as $\mathcal{I}_{\mathcal{T}, \mathcal{A}_q}$ (recall: \mathcal{A}_q is q viewed as an ABox, i.e., the variables in q are viewed as the ABox individuals of \mathcal{A}_q). The following is proved in [16].

Lemma 5.30. *For all CQs q, q' and DL-Lite TBoxes \mathcal{T} we have:*

- $q \sqsubseteq_{\mathcal{T}} q'$ iff $\mathcal{T}, \mathcal{A}_q \models q'$ iff $\mathcal{I}_{\mathcal{T}, q} \models q'$.

Using these auxiliary results, we can prove Lemma 5.28.

Proof (of Lemma 5.28). “ \Rightarrow ”: We make a case distinction on how a role R can be \mathcal{T} -generated in q .

- (i) $R(x, y) \sqsubseteq_{\mathcal{T}} q$. By Lemma 5.18, $R(x, y)$ must be a disjunct of $q_{\mathcal{T}}$ and we are done.
- (ii) $R(x, y) \in q$ with $y \neq x$ and x root variable. We claim that $q_{\mathcal{T}}$ has a disjunct that includes an atom $R(x', y')$ with x' a root variable. By Lemma 5.18, there is a disjunct p of $q_{\mathcal{T}}$ with $q \sqsubseteq p$ and $p \sqsubseteq_{\mathcal{T}} q$. If $p = q$, we are done. If not, let h be the homomorphism from p to q .

We first show that there is some atom $R(x', y') \in p$ with $h(x') = x$ and $h(y') = y$. Assume that this is not the case. Then h is also a homomorphism to $q' = q \setminus \{R(x, y)\}$. Hence $q' \sqsubseteq p$. As by assumption $p \sqsubseteq_{\mathcal{T}} q$, we obtain $q' \sqsubseteq_{\mathcal{T}} q$, contradicting \mathcal{T} -minimality of q .

Consider the atom $R(x', y')$ in p whose existence we just proved. Clearly, $x' \neq y'$ since $h(x') \neq h(y')$. Thus, it remains to show that x' is a root variable in p . Note that standard approaches such as the one in [28] generate UCQ-rewritings in which every disjunct is a simple tree query if q is a simple tree query. Hence, if x' is not a root variable in p , then y' is one. Thus, there must be an atom in p involving y' but not x' . However, as $h(y') = y$ and y is a fresh variable, the only atom involving y is $R(x, y)$. Thus, the atom **at** cannot exist, contradiction.

- (iii) $A(x) \in q$, x root variable in q , and $\mathcal{T} \models \exists R \sqsubseteq A$. Define $p = (q \setminus \{A(x)\}) \cup \{R(x, y)\}$ for some fresh variable y . Note that p is a query with the same properties as q under the previous point (ii). Thus, one can show in the same way that there is a disjunct p' in $q_{\mathcal{T}}$ that includes an atom $R(x', y')$ with x' a root variable.
- (iv) $S(x, y) \in q$, x root variable, $y \neq x$ occurs only once, and $\mathcal{T} \models \exists S \sqsubseteq \exists R$. Identical to previous case.

“ \Leftarrow ”: Assume now that $q_{\mathcal{T}}$ contains a disjunct p with an atom $R(x, y)$ where x is a root variable and $x \neq y$. We show that R is \mathcal{T} -generated in q . As $p \sqsubseteq_{\mathcal{T}} q$, there is a match π for q in $\mathcal{I}_{\mathcal{T}, p}$ by Lemma 5.30. It is convenient to view $\mathcal{I}_{\mathcal{T}, p}$ as a (potentially infinite) ABox $\mathcal{A}_{\mathcal{T}, p}$ that contains an assertion $A(a)$ for each $a \in A^{\mathcal{I}_{\mathcal{T}, p}}$ and an assertion $r(a, b)$ for each $(a, b) \in r^{\mathcal{I}_{\mathcal{T}, p}}$. By construction of the canonical model $\mathcal{I}_{\mathcal{T}, p}$, for every such assertion α we find a *single* atom $\text{cause}(\alpha) \in p$ that ‘causes’ α in the sense that α is part of the canonical model of $\{\text{cause}(\alpha)\}$ and \mathcal{T} . In case there are multiple possible choices for such a cause, we choose an arbitrary one.

For every atom **at** $\in q$, the match π identifies an assertion $\pi(\text{at}) \in \mathcal{A}_{\mathcal{T}, p}$. Let $p' \subseteq p$ denote the query

$$p' = \{\text{cause}(\pi(\text{at})) \mid \text{at} \in q\}.$$

By definition of p' , we have $\mathcal{I}_{\mathcal{T}, p'} \models q$ and thus $p' \sqsubseteq_{\mathcal{T}} q$. By Lemma 5.18, $p \sqsubseteq p'$, and the fact that p is a disjunct in $q_{\mathcal{T}}$, we get that $p \equiv p'$, thus $p = p'$. Hence, every atom in p is necessary.

Let us first consider the cases when p or q do not have unique root variables. If q has two root variables then it is of the form $S_1(x', y'), \dots, S_k(x', y')$ and we have $q_{\mathcal{T}} = q$. Thus, $p = q$ and q contains the mentioned atom $R(x, y)$ and R is \mathcal{T} -generated in q . If p has two root variables, then the following claim implies that R is \mathcal{T} -generated in q :

Claim. If p has two root variables, then either $p = R(x, y)$ or $p \equiv q$ and they are of the form $S_1(x, y), \dots, S_k(x, y)$ for $k \geq 2$.

Proof of Claim. Assume that p has two root variables, that is, $p = S_1(x, y), \dots, S_k(x, y)$ and $S_i = R$ for some i . If $k = 1$, we have $p = R(x, y)$. For $k > 1$, define

$$p'' = \{\text{cause}(\pi(\text{at})) \mid \text{cause}(\pi(\text{at})) = \pi(\text{at}), \text{at} \in q\}.$$

Construct a query \hat{p} as follows. Start with p'' and add for every $\text{at} \in q$ with $\text{cause}(\pi(\text{at})) \neq \pi(\text{at})$ the atoms $S(x, y')$, $S(x', y)$ where $S(x, y) = \text{cause}(\pi(\text{at}))$ and x', y' are fresh variables. By construction, π is a match for q in $\mathcal{I}_{\mathcal{T}, \hat{p}}$ and thus $\hat{p} \sqsubseteq_{\mathcal{T}} q$ by Lemma 5.30. Now note that there is a homomorphism from \hat{p} to p , that is, $p \sqsubseteq \hat{p}$. By Lemma 5.18, we have $p \equiv \hat{p}$. However, this implies that $p = p''$, thus π is a homomorphism from q to p , that is, $p \sqsubseteq q$. By Lemma 5.18 again, we get $p \equiv q$. This finishes the proof of the Claim. We can thus assume that the unique root variable of p is x and the unique root variable of q is x' . We make a case distinction on where π maps x' in $\mathcal{I}_{\mathcal{T}, p}$.

- Assume first $\pi(x') = a_x$. By what was said above, there is an atom $\text{at} \in q$ such that $\text{cause}(\pi(\text{at})) = R(x, y)$. If $\text{at} = R(x', y')$, we are done. If $\text{at} = A(x')$, the construction of $\mathcal{I}_{\mathcal{T}, p}$ yields that $\mathcal{T} \models \exists R \sqsubseteq A$, thus R is \mathcal{T} -generated in q . Assume now $\text{at} = S(x', y')$. Again, the construction of $\mathcal{I}_{\mathcal{T}, p}$ implies $\mathcal{T} \models \exists R \sqsubseteq \exists S$. Note that y' cannot appear twice in q as then $R(x, y)$ cannot be the ‘cause’ of at .
- So assume finally that $\pi(x') \neq a_x$, that is x' is mapped either to the anonymous part of $\mathcal{I}_{\mathcal{T}, p}$ or to some $a_z \in \text{Ind}(\mathcal{A}_p)$, $z \neq x$. As q is connected and a simple tree, there is a single $z \in \text{var}(q)$ such that π maps all variables to either a_x or elements of the form $a_z \cdot c_{R_1} \cdots c_{R_k}$. This variable z can only be y as otherwise, $R(x, y)$ is not necessary in p , contradiction. However, in this case all atoms in p that involve x but not y cannot be a cause of some atom in q , by construction of $\mathcal{I}_{\mathcal{T}, p}$. As x is the unique root variable, there is at least one such atom. Thus, p is not minimal, contradiction.

■

We are now ready to prove Theorem 5.26.

Proof (of Theorem 5.26). “ \Rightarrow ”: We show the contrapositive, that is, if q is not a simple tree query or none of the conditions (1) and (2) are satisfied, then $\text{ipOBDA}(q, \mathcal{T})$ is

#P-hard. If q is not a simple tree query, Theorem 5.25 implies that answering q relative to \mathcal{T} is #P-hard. Assume now that q is a simple tree query but conditions (1) and (2) are not satisfied, that is,

- (1') q is not of the form $S_1(x, y) \dots, S_k(x, y)$ for roles S_1, \dots, S_k , and
- (2') there is a role name r such that both r and r^- are \mathcal{T} -generated in q but $r(x, y) \not\sqsubseteq_{\mathcal{T}} q$.

In the proof of Lemma 5.28, we have already argued that we can write $q_{\mathcal{T}}$ as a disjunctive sentence with root variable x^* . By Lemma 5.28 and Item (2'), there are disjuncts q_1, q_2 of $q_{\mathcal{T}}$ and variables y_1, y_2 such that $r(x^*, y_1) \in q_1$, $y_1 \neq x^*$ and $r(y_2, x^*) \in q_2$, $y_2 \neq x^*$. Assume first some q_i has two root variables. In this case, we have shown (in the claim) in the proof of Lemma 5.28 that either $q_i = r(x, y)$ or $q_i = q$, where q is of the form $S_1(x, y), \dots, S_k(x, y)$ for some $k \geq 2$. However, the former is in contradiction to assumption (2') while the latter contradicts (1'). Thus, x^* is the unique root variable, but clearly it is no weak separator variable. By Theorem 5.19 $q_{\mathcal{T}}$ is #P-hard, thus $\text{ipOBDA}(q, \mathcal{T})$ is #P-hard.

“ \Leftarrow ”. Assume that q is a simple tree query that, together with \mathcal{T} , satisfies condition (1) or (2) from the theorem. In case of (1), $q_{\mathcal{T}} = q$ and $q_{\mathcal{T}}$ is in PTIME by Theorem 5.19. Otherwise, there is a unique root variable x^* . Assume now that there are two disjuncts of $q_{\mathcal{T}}$ that contain atoms $r(x^*, y)$ and $r(z, x^*)$ with $y \neq x^*$ and $z \neq x^*$. By Lemma 5.28, both r and r^- are \mathcal{T} -generated in q . By condition 2, we have $r(x, y) \sqsubseteq_{\mathcal{T}} q$. By Lemma 5.18, $r(x, y)$ is a disjunct in $q_{\mathcal{T}}$ which is in contradiction to the existence of the two disjuncts assumed above, that is, minimality of $q_{\mathcal{T}}$. ■

5.5 Beyond First-order Rewritings: \mathcal{ELI} -TBoxes

In the previous sections, we have established FO-rewritability as a tool for proving PTIME results for CQ answering in the context of probabilistic OBDA. The aim of this section is to establish that, in a sense, the tool is *complete*: we prove that whenever a CQ q is not FO-rewritable relative to a TBox \mathcal{T} , then q is #P-hard relative to \mathcal{T} ; thus, when a query is in PTIME relative to a TBox \mathcal{T} , then this can *always* be shown via FO-rewritability. To achieve this goal, we select a DL as the TBox language that, unlike DL-Lite, also embraces non FO-rewritable CQs/TBoxes. Here we choose \mathcal{ELI} , which properly generalizes DL-Lite. Note that, in traditional OBDA, there is a drastic difference in data complexity of CQ-answering between DL-Lite and \mathcal{ELI} : the former is in AC_0 while the latter is PTIME-complete.

In what follows, we again focus on Boolean, connected CQs q which can additionally involve individual names. Our main theorem in this section is as follows.

Theorem 5.31. *If a Boolean, connected CQ q is not FO-rewritable relative to an \mathcal{ELI} -TBox \mathcal{T} , then $\text{ipOBDA}(q, \mathcal{T})$ is #P-hard.*

As a consequence, we obtain an (abstract) dichotomy for full \mathcal{ELI} .

Theorem 5.32 (\mathcal{ELI} dichotomy). *Let q be a Boolean, connected CQ and \mathcal{T} an \mathcal{ELI} -TBox. Then q is in PTIME relative to \mathcal{T} or $\#P$ -hard relative to \mathcal{T} .*

Proof. If q is not FO-rewritable relative to \mathcal{T} , then it is $\#P$ -hard, by Theorem 5.31. Otherwise, we get a dichotomy by Theorem 5.12. ■

The proof of Theorem 5.31 is rather long and involves some technical parts. We start ‘at the tail’ and illustrate which property of *non*-FO-rewritability we are going to exploit. We call an ABox connected if the graph (V, E) given by $V = \text{Ind}(\mathcal{A})$ and $(a, b) \in E$ iff $r(a, b) \in \mathcal{A}$ for some r is connected.

Lemma 5.33. *If a Boolean connected CQ q is not FO-rewritable relative to an \mathcal{ELI} -TBox \mathcal{T} , then for every $d > 0$ there is a connected ABox $\widehat{\mathcal{A}}_d$ containing assertions*

$$R_1(a_1, a_2), \dots, R_d(a_d, a_{d+1})$$

such that

- a_1, \dots, a_{d+1} do not occur in q ;
- $\mathcal{T}, \widehat{\mathcal{A}}_d \models q$, but $\mathcal{T}, \mathcal{A}' \not\models q$ when \mathcal{A}' is $\widehat{\mathcal{A}}_d$ with any of those assertions dropped;
- dropping any of those assertions makes the ABox disconnected.

Having established Lemma 5.33, we can prove Theorem 5.31 by a reduction from the $\#P$ -hard problem MonBiDNF. More specifically, let φ be a formula of the form $(x_{i_1} \wedge y_{j_1}) \vee \dots \vee (x_{i_k} \wedge y_{j_k})$ where the set $X = \{x_1, \dots, x_{n_x}\}$ of variables that occur on the left-hand side of a conjunction in φ is disjoint from the set $Y = \{y_1, \dots, y_{n_y}\}$ of variables that occur on the right-hand side of a conjunction in φ . We define an ipABox $(\mathcal{A}_\varphi, p_\varphi)$ by starting with the ABox $\mathcal{A} := \widehat{\mathcal{A}}_3$ from Lemma 5.33 and duplicating the assertions $R_1(a_1, a_2)$, $R_2(a_2, a_3)$, $R_3(a_3, a_4)$ using fresh individual names b_1, \dots, b_{n_x} and c_1, \dots, c_{n_y} as follows.

- start with the ABox \mathcal{A} without $R_1(a_1, a_2)$, $R_2(a_2, a_3)$, $R_3(a_3, a_4)$; assign probability 1 to all assertions;
- add the following assertions with probability 1:

$$\begin{array}{ll} A(b_i) & \text{for all } A(a_2) \in \mathcal{A} \text{ and } 1 \leq i \leq n_x \\ R(b_i, d) & \text{for all } R(a_2, d) \in \mathcal{A} \text{ and } 1 \leq i \leq n_x \\ A(c_i) & \text{for all } A(a_3) \in \mathcal{A} \text{ and } 1 \leq i \leq n_y \\ R(c_i, d) & \text{for all } R(a_3, d) \in \mathcal{A} \text{ and } 1 \leq i \leq n_y \\ R_2(b_i, c_j) & \text{for each disjunct } x_i \wedge y_j \text{ in } \varphi; \end{array}$$

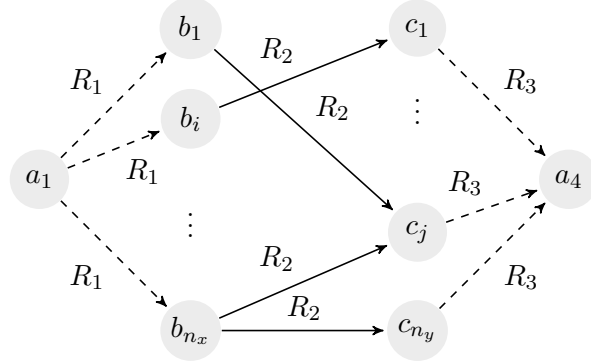


Figure 5.3: Gadget for the #P-hardness proof.

- add the following assertions with probability 0.5,

$$\begin{aligned} R_1(a_1, b_i) & \quad \text{for } 1 \leq i \leq n_x \\ R_3(c_i, a_4) & \quad \text{for } 1 \leq i \leq n_y. \end{aligned}$$

The construction is illustrated in Figure 5.3 where, in the middle part, there is an R_2 -edge from b_i to c_j if, and only if, $x_i \wedge y_j$ is a disjunct in φ , and dashed arrows indicate probabilistic assertions. Apart from what is shown in the figure, each b_i receives exactly the same concept and role assertions that a_2 has in \mathcal{A} , and each c_i is, in the same sense, a duplicate of a_1 in \mathcal{A} .

We are interested in ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\varphi$ with $p_\varphi(\mathcal{A}') > 0$. Each such ABox has probability $\frac{1}{2^{|X|+|Y|}}$ and corresponds to a truth assignment $\delta_{\mathcal{A}'}$ to the variables in $X \cup Y$: for $x_i \in X$, $\delta_{\mathcal{A}'}(x_i) = 1$ iff $R_1(a_1, b_i) \in \mathcal{A}'$ and for $y_i \in Y$, $\delta_{\mathcal{A}'}(y_i) = 1$ iff $R_3(c_i, a_4) \in \mathcal{A}'$. Let $\#\varphi$ the number of truth assignments to the variables $X \cup Y$ that satisfy φ . To complete the reduction, we show that $p(\mathcal{T}, \mathcal{A}_\varphi \models q) = \frac{\#\varphi}{2^{|X|+|Y|}}$. By what was said above, this is an immediate consequence of the following claim.

Claim. For all ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\varphi$ with $p_\varphi(\mathcal{A}') > 0$, $\delta_{\mathcal{A}'} \models \varphi$ iff $\mathcal{T}, \mathcal{A}' \models q$.

Proof of the Claim. “if”. Let $\delta_{\mathcal{A}'} \not\models \varphi$ and assume to the contrary of what is to be shown that $\mathcal{T}, \mathcal{A}' \models q$. Since $\delta_{\mathcal{A}'} \not\models \varphi$ and by construction of \mathcal{A}_φ , there are no i, j such that $R_1(a_1, b_i), R_2(b_i, c_j), R_3(c_j, a_4) \in \mathcal{A}'$, so in particular \mathcal{A}' decomposes into two components \mathcal{B}_1 and \mathcal{B}_2 where \mathcal{B}_1 contains the individual a_1 and \mathcal{B}_2 contains the individual a_4 . As q is connected and \mathcal{T} is an \mathcal{ELI} -TBox, we have either $\mathcal{T}, \mathcal{B}_1 \models q$ or $\mathcal{T}, \mathcal{B}_2 \models q$. Obtain \mathcal{B}'_1 and \mathcal{B}'_2 as follows:

- \mathcal{B}'_1 is the connected component of a_3 after dropping assertion $R_3(a_3, a_4)$ from \mathcal{A} (recall that \mathcal{A} becomes disconnected, when dropping one atom);
- \mathcal{B}'_2 is the connected component of a_2 after dropping assertion $R_1(a_1, a_2)$ from \mathcal{A} .

Clearly, \mathcal{B}_i can be embedded into \mathcal{B}'_i for both $i = 1, 2$ preserving the constants that appear in q .⁴ Thus, we also have that either $\mathcal{T}, \mathcal{B}'_1 \models q$ or $\mathcal{T}, \mathcal{B}'_2 \models q$, both contradicting Lemma 5.33.

“only if”. By construction of \mathcal{A}_φ , $\delta_{\mathcal{A}'} \models \varphi$ implies that, up to renaming of individual names that do not occur in q , we have $\mathcal{A} \subseteq \mathcal{A}'$. Since $\mathcal{T}, \mathcal{A} \models q$ by choice of \mathcal{A} , we must thus also have $\mathcal{T}, \mathcal{A}' \models q$.

This finishes the proof of the Claim and thus of Theorem 5.31. The rest of this section is devoted to the missing proof of Lemma 5.33. It consists of three steps:

- In the first step, we show that we can assume the TBox to be in a certain normal form.
- In the second step, we introduce an appropriate fixpoint operator and define that a query q is *bounded relative to a TBox* \mathcal{T} if the operator stabilizes after finitely many steps, similar to [102]. Boundedness is the central notion used here. In particular, we will show that non-FO-rewritability implies unboundedness.
- Finally, we identify for every $d > 0$ a suitable ABox \mathcal{A}_d and show, based on unboundedness, that \mathcal{A}_d can be transformed into an ABox having the properties from Lemma 5.33.

5.5.1 TBox Normalization

Let \mathcal{T} be an \mathcal{ELI} -TBox and q a Boolean connected CQ. We show that we can assume without loss of generality that \mathcal{T} contains only CIs of the forms

$$\begin{array}{ll} A \sqsubseteq B & A \sqsubseteq \exists R.B \\ B_1 \sqcap B_2 \sqsubseteq A & \exists R.B \sqsubseteq A \end{array}$$

where R ranges over roles and A, B, B_1, B_2 range over concept names and \top . Let $\text{sub}(\mathcal{T})$ denote the set of all subconcepts of (concepts that occur in) \mathcal{T} and reserve a concept name X_C for every $C \in \text{sub}(\mathcal{T}) \setminus (\mathbf{N}_C \cup \{\top\})$ such that X_C occurs neither in \mathcal{T} nor in q . Set

$$\sigma(C) = \begin{cases} C & \text{if } C \in \mathbf{N}_C \cup \{\top\}, \\ X_{D_1} \sqcap X_{D_2} & \text{if } C = D_1 \sqcap D_2, \\ \exists r.X_D & \text{if } C = \exists r.D. \end{cases}$$

Then put

$$\mathcal{T}' = \bigcup_{C \sqsubseteq D \in \mathcal{T}} X_C \sqsubseteq X_D \quad \cup \quad \bigcup_{C \in \text{sub}(\mathcal{T}) \setminus (\mathbf{N}_C \cup \{\top\})} X_C \equiv \sigma(C).$$

⁴We have not formally defined the *embedding* of an ABox into an other; we mean it in the sense of existence of a homomorphism on the level of assertions, i.e., individual names do not have to be preserved.

where $C \equiv D$ abbreviates $C \sqsubseteq D, D \sqsubseteq C$. After further replacing each CI of the form $A \sqsubseteq B_1 \sqcap B_2$ with $A \sqsubseteq B_1$ and $A \sqsubseteq B_2$, \mathcal{T}' is of the required syntactic form. Clearly, the conversion can be done in polynomial time.

We want to replace \mathcal{T} with the TBox \mathcal{T}' in normal form. To implement this, we consider ABoxes in a restricted signature. A *predicate* is either a concept name or a role name and a *signature* is a set of predicates. We use $\text{sig}(\mathcal{T})$ to denote the set of predicates that occur in \mathcal{T} and likewise for $\text{sig}(q)$. A Σ -ABox is an ABox that contains only symbols from Σ . We call a query q FO-rewritable relative to \mathcal{T} over Σ -ABoxes if there is a FOQ $q_{\mathcal{T}}$ such that $\text{cert}_{\mathcal{T}}(q, \mathcal{A}) = \text{ans}(q_{\mathcal{T}}, \mathcal{A})$ for all Σ -ABoxes. Due to the following result, we are indeed able to replace \mathcal{T} with \mathcal{T}' when we are careful about ABox signatures.

Theorem 5.34. *Let $\Sigma = \text{sig}(\mathcal{T}) \cup \text{sig}(q)$.*

1. *q is FO-rewritable relative to \mathcal{T} (over all ABoxes) iff q is FO-rewritable relative to \mathcal{T}' over Σ -ABoxes;*
2. *q is #P-hard relative to \mathcal{T} (over all ipABoxes) iff q is #P-hard relative to \mathcal{T}' over Σ -ipABoxes.*

Proof. For Point 1, first assume that q is FO-rewritable relative to \mathcal{T} and let φ be an FO-rewriting. We show that φ is also an FO-rewriting of q relative to \mathcal{T}' over Σ -ABoxes. To this end, let \mathcal{A} be a Σ -ABox. Since the fresh concept names X_C occur neither in q nor in \mathcal{A} , it is easy to show that $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{T}', \mathcal{A} \models q$. Since the former is equivalent to $\mathcal{I}_{\mathcal{A}} \models \varphi$, we are done. Conversely, assume that q is FO-rewritable relative to \mathcal{T}' over Σ -ABoxes and let φ' be an FO-rewriting. Since each non- Σ -symbol is interpreted as the empty set in $\mathcal{I}_{\mathcal{A}}$ for any Σ -ABox \mathcal{A} , we can w.l.o.g. assume that no such symbol occurs in φ' (if it does, replace it with **false**). We show that φ' is also an FO-rewriting of q relative to \mathcal{T} (over all ABoxes). Let \mathcal{A} be an ABox and $\mathcal{A}|_{\Sigma}$ the result of dropping all non- Σ -assertions from \mathcal{A} . We have:

$$\begin{array}{ll}
 \mathcal{T}, \mathcal{A} \models q \text{ iff } \mathcal{T}, \mathcal{A}|_{\Sigma} \models q & \text{since } q \text{ and } \mathcal{T} \text{ contain only } \Sigma\text{-symbols} \\
 \text{iff } \mathcal{T}', \mathcal{A}|_{\Sigma} \models q & \text{since the } X_C \text{ occur neither in } q \text{ nor in } \mathcal{A}|_{\Sigma} \\
 \text{iff } \mathcal{I}_{\mathcal{A}|_{\Sigma}} \models \varphi' & \text{since } \varphi' \text{ is an FO-rewriting} \\
 \text{iff } \mathcal{I}_{\mathcal{A}} \models \varphi' & \text{since there are no non-}\Sigma\text{-symbols in } \varphi'.
 \end{array}$$

For Point 2, first assume that q is #P-hard relative to \mathcal{T} . Since \mathcal{T} and q contain only Σ -symbols, we have that $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{T}, \mathcal{A}|_{\Sigma} \models q$ for all ABoxes \mathcal{A} and thus $p(\mathcal{T}, \mathcal{A} \models q) = p(\mathcal{T}, \mathcal{A}|_{\Sigma} \models q)$ for all ipABoxes \mathcal{A} . It follows that q is #P-hard relative to \mathcal{T} over Σ -ipABoxes. Since the fresh symbols X_C do not occur in \mathcal{A} or q , we have $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{T}', \mathcal{A} \models q$ for all Σ -ABoxes \mathcal{A} and thus also $p(\mathcal{T}, \mathcal{A} \models q) = p(\mathcal{T}', \mathcal{A} \models q)$ for all Σ -ipABoxes. Hence, answering q relative to \mathcal{T} over Σ -ipABoxes is simply the same problem as answering q relative to \mathcal{T}' over Σ -ipABoxes and we are done. For the converse

direction, $p(\mathcal{T}, \mathcal{A} \models q) = p(\mathcal{T}', \mathcal{A} \models q)$ for all Σ -ABoxes \mathcal{A} means that answering q relative to \mathcal{T}' over Σ -ipABoxes is simply a subproblem of answering q relative to \mathcal{T} , thus $\#P$ -hardness of the former implies $\#P$ -hardness of the latter. ■

From now on, we will assume that \mathcal{T} is in the normal form described above and denote with Σ the signature before establishing normal form. In particular, \mathcal{T} might contain non- Σ symbols and we will always take care that input ABoxes use only symbols from Σ .

5.5.2 Fixpoint operator, Boundedness, and FO-rewritability

The purpose of this part is to establish a connection between FO-rewritability and boundedness of an appropriate fixpoint operator, similar to what was observed in [102]. Given an ABox \mathcal{A} and an $a \in \text{Ind}(\mathcal{A})$, we denote with $\mathcal{A}|_a$ the *neighbourhood* of a , i.e., the restriction of \mathcal{A} to the individual name a and all members of $\{b \mid R(a, b) \in \mathcal{A}\}$ where R is a role name or its inverse. For a TBox \mathcal{T} , define

$$f_{\mathcal{T}}(\mathcal{A}) = \mathcal{A} \cup \{A(a) \mid a \in \text{Ind}(\mathcal{A}) \wedge \mathcal{T}, \mathcal{A}|_a \models A(a)\}.$$

Intuitively, $f_{\mathcal{T}}$ *locally* applies the TBox to its argument in order to materialize implicit knowledge. We set

$$f_{\mathcal{T}}^{\infty}(\mathcal{A}) := \bigcup_{i \geq 0} f_{\mathcal{T}}^i(\mathcal{A}),$$

where $f_{\mathcal{T}}^i(\cdot)$ denotes application of $f_{\mathcal{T}}$, iterated i times. Note that the application of $f_{\mathcal{T}}(\cdot)$ yields ABoxes, though not necessarily Σ -ABoxes. It is not hard to prove that for all $A \in \mathbf{N}_{\mathcal{C}}$ and $a \in \mathbf{N}_{\mathcal{I}}$, we have $\mathcal{T}, \mathcal{A} \models A(a)$ iff $A(a) \in f_{\mathcal{T}}^{\infty}(\mathcal{A})$ [102]. We want to establish an analogous claim for CQs, which requires first introducing several technical notions. In particular, we construct a query $q_{\mathcal{T}}$ from q and \mathcal{T} such that answering q over a Σ -ABox \mathcal{A} is equivalent to answering $q_{\mathcal{T}}$ over $f_{\mathcal{T}}^{\infty}(\mathcal{A})$. Note that, despite reusing the symbol, $q_{\mathcal{T}}$ is in general *not* an FO-rewriting of q relative to \mathcal{T} , which are not guaranteed to exist in \mathcal{ELI} ; intuitively, $q_{\mathcal{T}}$ constitutes the part of the FO-rewriting that deals with objects generated by existential quantifiers.

We use $\text{EQ}(q)$ to denote the set of all equivalence relations on $\text{term}(q)$ such that $a \not\sim b$ for all distinct $a, b \in \mathbf{N}_{\mathcal{I}}$. For every $\sim \in \text{EQ}(q)$, define a *collapsing* q_{\sim} of q :

$$q_{\sim} = \{r(s_{[t]}, s_{[t']}) \mid r(t, t') \in q\} \cup \{A(s_{[t]}) \mid A(t) \in q\}.$$

where $s_{[t]} = a$ if the individual name a is in $[t]$ and $s_{[t]}$ is the fresh variable $z_{[t]}$ otherwise. We call a CQ q *tree-shaped* if there no cycle in q , that is, there is no sequence of distinct atoms $R_1(t_1, t_2), R_2(t_2, t_3), \dots, R_k(t_k, t_{k+1})$ in q with $t_{k+1} = t_1$. A *splitting* S of a CQ q is a partition q_0, \dots, q_k of q (with q_0 possibly empty and q_1, \dots, q_k non-empty) such that we have

1. q_1, \dots, q_k are tree-shaped queries with roots t_1, \dots, t_k ;
2. $\text{term}(q_i) \cap \text{term}(q_j) = \emptyset$ for $1 \leq i < j \leq k$;
3. $\text{term}(q_0) \cap \text{term}(q_i) = \{t_i\}$ for $1 \leq i \leq k$;
4. Each q_i does not contain individual names except for possibly t_i .

Let $\text{split}(q)$ denote the set of all splittings of q and let $\text{CN}(\mathcal{T})$ denote the set of all concept names occurring in \mathcal{T} . Fix a splitting $S = q_0, \dots, q_k$ of a CQ q and a map $\rho : \{1, \dots, k\} \rightarrow 2^{\text{CN}(\mathcal{T})}$. We use $\mathcal{A}_{\rho,i}$ to denote the ABox $\{A(\hat{t}_i) \mid A \in \rho(i)\}$ where $\hat{t}_i = t_i$ if t_i is an individual name and a fresh individual name \hat{a} otherwise. We say that ρ is a *justification for S relative to \mathcal{T}* if for all $i \in \{1, \dots, k\}$, we have $\mathcal{T}, \mathcal{A}_{\rho,i} \models^{t_i} q_i$ meaning that in every model \mathcal{I} of $\mathcal{A}_{\rho,i}$ and \mathcal{T} , there is a match of the tree-shaped Boolean query q_i that maps its root t_i to $\hat{t}_i^{\mathcal{I}}$. Let $\text{just}(S, \mathcal{T})$ denote all possible justifications of S relative to \mathcal{T} . We use $q_{\rho(i)}$ to denote the query $\bigwedge_{A \in \rho(i)} A(t_i)$ where, again t_i denotes the root of q_i . For each $\sim \in \text{EQ}(q)$, set

$$\hat{q}_{\sim} = \bigvee_{S=q_0, \dots, q_k \in \text{split}(q_{\sim})} \exists z_{[\hat{t}_1]} \cdots \exists z_{[\hat{t}_n]} \bigvee_{\rho \in \text{just}(S, \mathcal{T})} (q_0 \wedge \bigwedge_{i=1}^k q_{\rho(i)})$$

where $[\hat{t}_1], \dots, [\hat{t}_n]$ are the equivalence classes of \sim on $\text{var}(q_0)$ that do not contain an individual name. Finally, we define the UCQ $q_{\mathcal{T}}$ as $\bigvee_{\sim \in \text{EQ}(q)} \hat{q}_{\sim}$.

Theorem 5.35. *For any CQ q , \mathcal{ELI} -TBox \mathcal{T} in normal form, and Σ -ABox \mathcal{A} , we have $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{I}_{f^{\infty}(\mathcal{A})} \models q_{\mathcal{T}}$.*

To prove Theorem 5.35, we introduce canonical models for ABoxes and \mathcal{ELI} -TBoxes. Let \mathcal{A} be an ABox and \mathcal{T} an \mathcal{ELI} -TBox in normal form. A *type for \mathcal{T}* is a set $T \subseteq \text{CN}(\mathcal{T})$. When $a \in \text{Ind}(\mathcal{A})$, T, T' are types for \mathcal{T} , and R is a role, we write

- $a \rightsquigarrow_R T$ if $\mathcal{T}, \mathcal{A} \models \exists R. \sqcap T(a)$ and for each type S for \mathcal{T} with $T \subsetneq S$, we have $\mathcal{T}, \mathcal{A} \not\models \exists R. \sqcap S(a)$;
- $T \rightsquigarrow_R T'$ if $\mathcal{T} \models \sqcap T \sqsubseteq \exists R. \sqcap T'$ and for each type S for \mathcal{T} with $T' \subsetneq S$, we have $\mathcal{T} \not\models \sqcap T \sqsubseteq \exists R. \sqcap S$.

A *path for \mathcal{A} and \mathcal{T}* is a sequence $p = aR_1T_1 \cdots T_{n-1}R_nT_n$, $n \geq 0$, with $a \in \text{Ind}(\mathcal{A})$, R_1, \dots, R_n roles, and T_1, \dots, T_n types for \mathcal{T} such that $a \rightsquigarrow_{R_1} T_1$ and $T_i \rightsquigarrow_{R_i} T_{i+1}$ for $1 \leq i < n$. When $n > 0$, we use $\text{tail}(p)$ to denote T_n . Let Paths be the set of all paths for

\mathcal{A} and \mathcal{T} and define the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ as follows:

$$\begin{aligned} \Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \text{Paths} \\ A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{a \in \text{Ind}(\mathcal{A}) \mid \mathcal{T}, \mathcal{A} \models A(a)\} \cup \\ &\quad \{p \in \Delta^{\mathcal{T}} \setminus \text{Ind}(\mathcal{A}) \mid \mathcal{T} \models \bigwedge \text{tail}(p) \sqsubseteq A\} \\ r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{(a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid r(a, b) \in \mathcal{A}\} \cup \\ &\quad \{(p, prT) \mid prT \in \text{Paths}\} \cup \\ &\quad \{(pr^-T, p) \mid pr^-T \in \text{Paths}\} \\ a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= a \end{aligned}$$

It is standard to prove that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is canonical in the following sense.

Lemma 5.36. *For any CQ q , \mathcal{ELI} -TBox \mathcal{T} in normal form, and ABox \mathcal{A} , we have $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q$.*

We are now ready to prove Theorem 5.35.

Proof (of Theorem 5.35). “if”. Assume that $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})} \models q_{\mathcal{T}}$. Thus, there is a disjunct q' of $q_{\mathcal{T}}$ with $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})} \models q'$. Let π be a match of q' in $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})}$. By definition of $q_{\mathcal{T}}$, there is an equivalence relation $\sim \in \text{EQ}$, a splitting $S = q_0, \dots, q_k \in \text{split}(q_{\sim})$ and a $\rho \in \text{just}(S, \mathcal{T})$ such that $q' = q_0 \wedge \bigwedge_{i=1}^k q_{\rho(i)}$. Note that $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})}$ is in fact the same as $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ restricted to $\text{Ind}(\mathcal{A})$. Thus, π is a match of q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ that maps only to $\text{Ind}(\mathcal{A})$. In particular, we have $\pi(t_i) \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ for every $A(t_i) \in q_{\rho(i)}$. As ρ is a justification for S relative to \mathcal{T} , we have $\mathcal{T}, \mathcal{A}_{\rho,i} \models^{t_i} q_i$, for $1 \leq i \leq k$. By construction of the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, there is thus a match π_i of q_i in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ such that the root t_i of q_i is mapped to $\pi(t_i)$. Define $\pi_0 : \text{term}(q_0) \rightarrow \text{Ind}(\mathcal{A})$ as follows:

$$\pi_0(t_i) = \begin{cases} a & \text{if } t \sim a \text{ for some } a \in \text{Ind}(\mathcal{A}), \\ \pi([t]) & \text{otherwise.} \end{cases}$$

It can be verified that $\pi_0 \cup \pi_1 \cup \dots \cup \pi_k$ is a match of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. By Lemma 5.36, we have $\mathcal{A}, \mathcal{T} \models q$ as required.

“only if”. Assume that $\mathcal{T}, \mathcal{A} \models q$. By Lemma 5.36, this means $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q$. Let π be a match of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ and let a_1, \dots, a_k be the elements of $\text{Ind}(\mathcal{A})$ that are in the range of π . Define a splitting $S = q_0, \dots, q_k$ of q , where

- q_0 involves all atoms $A(t), r(t, t')$ in q with $\pi(t), \pi(t') \in \text{Ind}(\mathcal{A})$;
- q_i involves all atoms $A(t), r(t, t')$ in $q \setminus q_0$ with $\pi(t), \pi(t')$ of the form $a_i p$ for some p .

Define a justification ρ for S relative to \mathcal{T} by setting $\rho(i) = \{A \in \text{CN}(\mathcal{T}) \mid a_i \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}\}$ for $1 \leq i \leq k$. Then, $q' = q_0 \wedge \bigwedge_{i=1}^k q_{\rho(i)}$ is a disjunct in $q_{\mathcal{T}}$ and it can be verified that $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})} \models q'$. \blacksquare

We will now provide the central definition of boundedness and its connection to (non-)FO-rewritability.

Definition 5.37 (Boundedness). A CQ q is k -bounded relative to a TBox \mathcal{T} over Σ -ABoxes if for every Σ -ABox \mathcal{A} , we have that $\mathcal{I}_{f_{\mathcal{T}}^k(\mathcal{A})} \models q_{\mathcal{T}}$ iff $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A})} \models q_{\mathcal{T}}$. We say that q is bounded relative to a TBox \mathcal{T} over Σ -ABoxes if it is k -bounded for some k .

Theorem 5.38. If a CQ q is bounded relative to \mathcal{T} over Σ -ABoxes, then it is FO-rewritable relative to \mathcal{T} over Σ -ABoxes.

Proof. Assume that q is bounded relative to \mathcal{T} over Σ -ABoxes and let $k > 0$ be such that $\mathcal{I}_{f_{\mathcal{T}}^k(\mathcal{A})} \models q_{\mathcal{T}}$ iff $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A})} \models q_{\mathcal{T}}$ for every Σ -ABox \mathcal{A} . The proof idea is to express the k -fold application of $f_{\mathcal{T}}$ in first-order logic, that is, we can give a first-order rewriting of q relative to \mathcal{T} .

Observe that Σ is finite, and thus there are only finitely many neighborhoods \mathcal{A}_a that can occur in a Σ -ABox \mathcal{A} , up to isomorphism. Every such neighborhood \mathcal{N} with individual name a in the center can be converted in a straightforward way into an existential, conjunctive, positive FO-formula:

$$\varphi_{\mathcal{N}} = \bigwedge_{A(a) \in \mathcal{N}} A(x) \wedge \bigwedge_{b \in \text{Ind}(\mathcal{N})} \exists y : \left(\bigwedge_{R(a,b) \in \mathcal{N}} R(x, y) \wedge \bigwedge_{B(b) \in \mathcal{N}} B(y) \right).$$

For each concept name A , we use Γ_A to denote the set of neighborhoods \mathcal{N} with center a such that $A(a) \in f_{\mathcal{T}}(\mathcal{N})$. For every concept name A and $i \geq 0$, set

- $q_A^0(x) := A(x)$
- $q_A^{i+1}(x) := q_A^i \vee \bigvee_{\mathcal{N} \in \Gamma_A} \varphi'_{\mathcal{N}}$ where $\varphi'_{\mathcal{N}}$ is obtained from $\varphi_{\mathcal{N}}$ by replacing, for each concept name B , every atom $B(z)$ with $q_B^i[z/x]$.

The following can be proved by induction on i :

Claim. For every Σ -ABox \mathcal{A} and $i \geq 0$, we have $\mathcal{I}_{\mathcal{A}} \models q_A^i[a]$ iff $A(a) \in f_{\mathcal{T}}^i(\mathcal{A})$.

Let $\hat{q}_{\mathcal{T}}$ be $q_{\mathcal{T}}$ with every atom $A(t)$ replaced with $q_A^k[t/x]$. We show that $\hat{q}_{\mathcal{T}}$ is an FO-rewriting of q relative to \mathcal{T} , which finishes the proof. By the above claim, we have $\mathcal{I}_{\mathcal{A}} \models \hat{q}_{\mathcal{T}}$ iff $\mathcal{I}_{f_{\mathcal{T}}^k(\mathcal{A})} \models q_{\mathcal{T}}$. By choice of k , this is the case iff $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A})} \models q_{\mathcal{T}}$. By Theorem 5.35, this is equivalent to $\mathcal{T}, \mathcal{A} \models q$. ■

5.5.3 Construction of $\widehat{\mathcal{A}}_d$ from Lemma 5.33

In the following, we fix some d and construct an ABox satisfying the properties of $\widehat{\mathcal{A}}_d$ from Lemma 5.33. Let q be unbounded relative to \mathcal{T} over Σ -ABoxes and set

$$m = |\mathcal{T}| \cdot |q_{\mathcal{T}}| \cdot (|\mathcal{T}| + |q_{\mathcal{T}}|)^d + 1.$$

The choice of m will become clear at the end of this section. Since q is not bounded relative to \mathcal{T} , there is a Σ -ABox \mathcal{A}_0 such that $\mathcal{I}_{f_{\mathcal{T}}^m}(\mathcal{A}_0) \models q_{\mathcal{T}}$, but $\mathcal{I}_{f_{\mathcal{T}}^{m-1}}(\mathcal{A}_0) \not\models q_{\mathcal{T}}$. As it is now, we cannot make any assumption about the shape of \mathcal{A}_0 . The goal here is to construct a ‘forest-shaped’ ABox with bounded outdegree with the same properties as \mathcal{A}_0 . In the first step, we show what is meant with ‘forest-shaped’.

Choose a concrete match π of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^m}(\mathcal{A}_0)$ and denote with $\text{ran}(\pi)$ the *range* of π , that is, the set $\{\pi(t) \mid t \in \text{term}(q_{\mathcal{T}})\}$. In the first step, we to construct a Σ -ABox \mathcal{A}_1 as follows:

- a π -path in \mathcal{A}_0 of length n is a sequence $p = a_0 R_1 a_1 \cdots R_n a_n$ such that $a_0 \in \text{ran}(\pi)$ and $R_i(a_{i-1}, a_i) \in \mathcal{A}_0$ for $1 \leq i \leq n$; we use $\text{tail}(p)$ to denote a_n ;
- $\text{Ind}(\mathcal{A}_1)$ is the set of all π -paths in \mathcal{A}_0 of length at most m ;
- if $A(a) \in \mathcal{A}_0$, $p \in \text{Ind}(\mathcal{A}_1)$, and $\text{tail}(p) = a$, then $A(p) \in \mathcal{A}_1$;
- if $pRa \in \text{Ind}(\mathcal{A}_1)$, then $R(p, pRa) \in \mathcal{A}_1$;
- if $r(a, b) \in \mathcal{A}_0$ and $a, b \in \text{ran}(\pi)$, then $r(a, b) \in \mathcal{A}_1$.
- these are all assertions in \mathcal{A}_1 .

Intuitively, \mathcal{A}_1 is a (finite) unraveling of \mathcal{A}_0 that preserves the range of π and is of depth m . Thus, \mathcal{A}_1 is of a special shape:

- the elements in $\text{ran}(\pi)$ form a ‘core’ whose relational structure is not restricted in any way;
- each ‘root’, that is, each element $a \in \text{ran}(\pi)$, gives rise to a tree-shaped sub-ABox of \mathcal{A}_1 , namely the restriction to the individuals p of the form $a \cdot p'$.

Lemma 5.39. $\mathcal{I}_{f_{\mathcal{T}}^m}(\mathcal{A}_1) \models q_{\mathcal{T}}$, but $\mathcal{I}_{f_{\mathcal{T}}^{m-1}}(\mathcal{A}_1) \not\models q_{\mathcal{T}}$.

Proof. We first verify the following claim.

Claim. For all $i \leq m$, $p \in \text{Ind}(\mathcal{A}_1)$ with $\text{tail}(p) = a$, and $A \in \mathbf{N}_{\mathcal{C}}$, we have

- (a) $A(p) \in f_{\mathcal{T}}^i(\mathcal{A}_1)$ implies $A(a) \in f_{\mathcal{T}}^i(\mathcal{A}_0)$.
- (b) if p is of length at most $m - i$, then $A(a) \in f_{\mathcal{T}}^i(\mathcal{A}_0)$ implies $A(p) \in f_{\mathcal{T}}^i(\mathcal{A}_1)$.

Proof of the Claim. The proof is by induction on i . The induction start is trivial by construction of \mathcal{A}_1 . For the induction step, note that, by definition of $f_{\mathcal{T}}$, we have $A(p) \in f_{\mathcal{T}}^i(\mathcal{A}_1)$ iff $f_{\mathcal{T}}^{i-1}(\mathcal{A}_1)|_p, \mathcal{T} \models A(p)$. By induction hypothesis and the construction

of \mathcal{A}_1 , we have $f_{\mathcal{T}}^{i-1}(\mathcal{A}_1)|_p \subseteq f_{\mathcal{T}}^{i-1}(\mathcal{A}_0)|_a$ with equality if p is of length at most $m - i$.⁵ This implies $f_{\mathcal{T}}^{i-1}(\mathcal{A}_0)|_a, \mathcal{T} \models A(a)$, i.e., $A(a) \in f_{\mathcal{T}}^i(\mathcal{A}_0)$, and implication in the converse direction holds, when the length of p is at most $m - i$. This finishes the proof of the Claim.

To see that $\mathcal{I}_{f_{\mathcal{T}}^m(\mathcal{A}_1)} \models q_{\mathcal{T}}$, let q' be a disjunct of $q_{\mathcal{T}}$ such that π (the match used for the construction of \mathcal{A}_1) is a match for q' in $\mathcal{I}_{f_{\mathcal{T}}^m(\mathcal{A}_0)}$. We claim that π is also a match of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^m(\mathcal{A}_1)}$:

- $r(t, t') \in q'$ implies $r(\pi(t), \pi(t')) \in f_{\mathcal{T}}^m(\mathcal{A}_0)$ and thus $r(\pi(t), \pi(t')) \in \mathcal{A}_0$. By construction of \mathcal{A}_1 , $r(\pi(t), \pi(t')) \in \mathcal{A}_1 \subseteq f_{\mathcal{T}}^m(\mathcal{A}_1)$;
- $A(t) \in q'$ implies $A(\pi(t)) \in f_{\mathcal{T}}^m(\mathcal{A}_0)$. By part (b) of the claim, $A(\pi(t)) \in f_{\mathcal{T}}^m(\mathcal{A}_1)$.

For $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)} \not\models q_{\mathcal{T}}$, assume that there is a match τ of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)}$. By part (a) of the claim and the construction of \mathcal{A}_1 , the mapping h that maps every element p of $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)}$ to $\text{tail}(p)$ in $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_0)}$ is a homomorphism. Thus, $h \circ \tau$ is a match for $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_0)}$, contradiction. ■

Next, we ‘minimize’ \mathcal{A}_1 by exhaustively applying the following operation: if $\alpha \in \mathcal{A}_1$ is an assertion involving a non-root element, that is, a path p of length at least 1, and $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_1^-)} \models q_{\mathcal{T}}$ for $\mathcal{A}_1^- = \mathcal{A}_1 \setminus \{\alpha\}$, then replace \mathcal{A}_1 with \mathcal{A}_1^- . Let \mathcal{A}_2 be the Σ -ABox finally obtained. For $i \geq 0$, we say that \mathcal{A}_2 *has outdegree at most i* if every individual in \mathcal{A}_2 has at most i successors that are not roots, that is, for every $p \in \text{Ind}(\mathcal{A}_1)$, we have

$$|\{p' \in \text{Ind}(\mathcal{A}_2) \setminus \text{Ind}(\mathcal{A}_0) \mid \exists R : R(p, p') \in \mathcal{A}_2\}| \leq i.$$

Lemma 5.40. \mathcal{A}_2 satisfies the following:

1. There is some $m' \geq m$ with $\mathcal{I}_{f_{\mathcal{T}}^{m'}(\mathcal{A}_2)} \models q_{\mathcal{T}}$, but $\mathcal{I}_{f_{\mathcal{T}}^{m'-1}(\mathcal{A}_2)} \not\models q_{\mathcal{T}}$;
2. \mathcal{A}_2 has outdegree at most $|\mathcal{T}| + |q_{\mathcal{T}}|$.

Proof. For Point 1, it suffices to show that $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_2)} \not\models q_{\mathcal{T}}$ and $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_2)} \models q_{\mathcal{T}}$. The former is a consequence of the facts that $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)} \not\models q_{\mathcal{T}}$, $\mathcal{A}_2 \subseteq \mathcal{A}_1$, and $q_{\mathcal{T}}$ is a UCQ; the latter is immediate by construction of \mathcal{A}_2 .

For Point 2, assume to the contrary of what is to be shown that there is a $p \in \text{Ind}(\mathcal{A}_2)$ such that the cardinality of

$$\Gamma = \{p' \in \text{Ind}(\mathcal{A}_2) \setminus \text{Ind}(\mathcal{A}_0) \mid \exists R : R(p, p') \in \mathcal{A}_2\}$$

exceeds $|\mathcal{T}| + |q_{\mathcal{T}}|$. Fix a match π of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_2)}$ and mark all those individual names in Γ that are in the range of π . For each $\exists R.A \sqsubseteq B \in \mathcal{T}$ such that $B(p) \in f_{\mathcal{T}}^\infty(\mathcal{A}_2)$, mark an individual name $p' \in \Gamma$ such that

⁵Containment of neighborhoods is meant independent from the individual names; for instance, $\{A(a), r(a, b)\} \subseteq \{A(c), r(c, d), B(d)\}$.

1. $R(p, p') \in \mathcal{A}_2$ and $A(p') \in f_{\mathcal{T}}^j(\mathcal{A})$ for some $j < m$;
2. there is no p'' that satisfies Point 1 for some smaller j .

Note that such a p' need not exist, in which case no node is marked for $\exists r.A \sqsubseteq B$. Since $|\Gamma| > |\mathcal{T}| + |q_{\mathcal{T}}|$, there is at least one element $p_0 \in \Gamma$ that is not marked. Consider the ABox \mathcal{A}_2^- obtained from \mathcal{A}_2 by dropping the (unique) assertion $R(p, p_0)$ that caused p_0 to be in Γ and all other assertions involving an individual name with prefix p_0 . As each disjunct in $q_{\mathcal{T}}$ is connected, we do not drop marked individuals. Based on the normal form of \mathcal{T} and the definition of $f_{\mathcal{T}}$, it is not hard to show that no deductions are lost.

Claim. For all $i \geq 0$, $p \in \text{Ind}(\mathcal{A}_2^-)$, and concept names A , we have $A(p) \in f_{\mathcal{T}}^i(\mathcal{A}_2^-)$ iff $A(p) \in f_{\mathcal{T}}^i(\mathcal{A}_2)$.

Proof of the Claim. The direction “only if” is immediate as $\mathcal{A}_2^- \subseteq \mathcal{A}_2$. For the other direction, observe that it is trivially true for $i = 0$. For $i > 0$, assume that $A(p)$ is added to $f_{\mathcal{T}}^i(\mathcal{A}_2)$. By definition of $f_{\mathcal{T}}$ and the normal form of \mathcal{T} , there is a point p' such that $R(p, p') \in \mathcal{A}_2$, $B(p') \in f_{\mathcal{T}}^{i-1}(\mathcal{A}_2)$, and $\exists R.B \sqsubseteq A \in \mathcal{T}$. If there is such a point $p' \neq p_0$, then p' is an individual name in \mathcal{A}_2^- and we can apply induction hypothesis. Otherwise, we have $p' = p_0$, contradicting the marking condition. This finishes the proof of the Claim.

Consequently, π is still a match of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_2^-)}$, contradicting minimality of \mathcal{A}_2 . ■

We are finally ready to prove that \mathcal{A}_2 satisfies the conditions of Lemma 5.33. We say that a path $p' \in \text{Ind}(\mathcal{A}_2)$ is an *extension* of a path $p \in \text{Ind}(\mathcal{A}_2)$ if $p' = pRa$ for some R and a . We claim that \mathcal{A}_2 contains a sequence of assertions

$$R_1(p_1, p_2), \dots, R_d(p_d, p_{d+1})$$

such that p_1 is of length one and p_{i+1} is an extension of p_i for all $i < d$. Assume this is not the case. Then all paths in $\text{Ind}(\mathcal{A})$ are of length at most d . Since \mathcal{A}_2 is forest-shaped with at most $|q_{\mathcal{T}}|$ roots, Point 2 of Lemma 5.40 yields $|\text{Ind}(\mathcal{A}_2)| \leq |q_{\mathcal{T}}| \cdot (|\mathcal{T}| + |q_{\mathcal{T}}|)^d$. Note that iterated applications of $f_{\mathcal{T}}$ can assign, in the worst case, every concept name from \mathcal{T} to every individual name. Hence, for an ABox \mathcal{B} with i elements, we have $f_{\mathcal{T}}^\ell(\mathcal{B}) = f_{\mathcal{T}}^{\ell+1}(\mathcal{B})$ where $\ell = |\mathcal{T}| \cdot i$. Applied to \mathcal{A}_2 , we obtain $f_{\mathcal{T}}^\ell(\mathcal{A}_2) = f_{\mathcal{T}}^{\ell+1}(\mathcal{A}_2)$ where $\ell = |\mathcal{T}| \cdot |q_{\mathcal{T}}| \cdot (|\mathcal{T}| + |q_{\mathcal{T}}|)^d$. This is in contradiction to Point 1 of Lemma 5.40 and the fact that $m' \geq m > \ell$.

It suffices to note that, by Theorem 5.35, $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_2)} \models q_{\mathcal{T}}$ implies $\mathcal{T}, \mathcal{A} \models q$. On the other hand, by construction of \mathcal{A}_2 , we have $\mathcal{I}_{f_{\mathcal{T}}^\infty(\mathcal{A}_2^-)} \not\models q_{\mathcal{T}}$ for every subset $\mathcal{A}_2^- = \mathcal{A}_2 \setminus \{R_i(p_i, p_{i+1})\}$. By Theorem 5.35, we get $\mathcal{T}, \mathcal{A}_2^- \not\models q$ for each such \mathcal{A}_2^- . Finally, by construction of \mathcal{A}_2 , the sequence is the unique path from p_1 to p_{d+1} .

5.6 Monte Carlo Approximation

The results in Sections 5.4 and 5.5 show that PTIME complexity is an elusive property even for ipABoxes and relatively inexpressive TBox languages such as DL-Lite and \mathcal{ELI} . Of course, the same is true for probabilistic databases, even for very simple data models such as tuple independent databases. To address this fundamental problem, researchers and users are often trading accuracy for efficiency, replacing exact answers with approximate ones. In particular, it is popular to use Monte Carlo approximation in the form of *fully polynomial randomized approximation schemes (FPRASes)*. An *FPRAS* for a Boolean CQ q and TBox \mathcal{T} is a randomized polynomial time algorithm that, given an ipABox \mathcal{A} and an error bound $\epsilon > 0$, computes a real number x such that

$$\Pr\left(\frac{|p(\mathcal{T}, \mathcal{A} \models q) - x|}{p(\mathcal{T}, \mathcal{A} \models q)} \leq \frac{1}{\epsilon}\right) \geq \frac{3}{4}.$$

In words: with a high probability—the value of $\frac{3}{4}$ can be amplified by standard methods—the algorithm computes a result that deviates from the actual result by at most factor $\frac{1}{\epsilon}$. The term ‘fully polynomial’ refers to the fact that the run-time of the algorithm is polynomial in *both* the size of \mathcal{A} and the error bound ϵ . For more details on FPRASes and related results, consult [128].

The goal of this section is revisit probabilistic OBDA in the light of FPRASes. In particular, we want to reconsider the results obtained so far and check the existence of an FPRAS in each case. A technical peculiarity is that *non-existence* of an FPRAS cannot be proved (so far) without complexity theoretic assumptions. For example, it is well-known that $\#\text{SAT}$ —and, in general, the counting version of any NP-hard decision problem—has no FPRAS unless $\text{NP} = \text{RP}$ [83], which is commonly assumed not to be the case. On the positive side, by a classical result by Karp and Luby, there is an FPRAS for $\#\text{DNF}$, that is, for counting the number of satisfying assignments of a propositional formula in DNF [88].

Let us start with considerations regarding our data model. It follows from the proof of Theorem 5.6 and the fact that $\#\text{SAT}$ does not admit an FPRAS that *over pABoxes* every CQ does not have an FPRAS relative to every FO-TBox. However, the existence of an FPRAS is not precluded per se in case the event expressions are restricted to DNF formulas (in contrast to $\#\text{P}$ -hardness of exact reasoning). Quite to the contrary, Fagin *et al.* observed that there is an FPRAS for UCQs over probabilistic databases when the event expressions are given in DNF [44, Theorem A.14]. We thus have:

Theorem 5.41. *Let q be an arbitrary UCQ. Then:*

- *There is no FPRAS for answering q relative to any FO-TBox over full pABoxes (unless $\text{RP} = \text{NP}$);*

- *There is an FPRAS for answering q relative to the empty TBox over pABoxes where event expressions are given in DNF.*

Motivated by the second point of the theorem, we turn our attention to restricted versions of pABoxes. Remarkably, FO-rewritability and the connection to tuple-independent databases turn out to be useful tools again. To be more specific, assume any pair of first-order query and TBox (φ, \mathcal{T}) such that φ is FO-rewritable relative to \mathcal{T} . By Proposition 5.15, there is a UCQ-rewriting $\varphi_{\mathcal{T}}$ of φ relative to \mathcal{T} . By the Lifting Theorem 5.13, we can answer $\varphi_{\mathcal{T}}$ over \mathcal{A} (viewed as a tuple-independent database) instead of computing $p(\mathcal{T}, \mathcal{A} \models \varphi)$ directly. By the mentioned observation from [44], there exists an FPRAS for $\varphi_{\mathcal{T}}$. We thus obtain:

Theorem 5.42. *If a FOQ φ is FO-rewritable relative to a FO-TBox \mathcal{T} , then there is an FPRAS for computing the query probability over pABoxes when event expressions are given in DNF. In particular, there is an FPRAS for ipOBDA(φ, \mathcal{T}).*

This observation clearly gives hope for the practical feasibility of probabilistic OBDA. In particular, Theorem 5.42 implies that for every CQ (even every UCQ) and DL-Lite-TBox, we can approximate the query probability. It is thus natural to ask whether FPRASes also exist for (CQs and) TBoxes formulated in richer ontology languages. No general positive result can be expected for expressive DLs that involve all Boolean operators such as \mathcal{ALC} . As analyzed in detail in [102], there is a large class of Boolean CQs q and \mathcal{ALC} -TBoxes \mathcal{T} such that, given a non-probabilistic ABox \mathcal{A} , it is coNP-hard to check the entailment $\mathcal{T}, \mathcal{A} \models q$. As argued above, the corresponding counting (or probability computation) problem cannot have an FPRAS, and thus we obtain the following.

Theorem 5.43. *There are CQs q and \mathcal{ALC} -TBoxes \mathcal{T} such that there is no FPRAS for q and \mathcal{T} (unless $\text{RP} = \text{NP}$).*

Consequently, it is interesting to study \mathcal{ELI} as the TBox language, where traditional OBDA is in PTIME data complexity for all CQs q and TBoxes \mathcal{T} . By what was said above, the case when q is FO-rewritable relative to \mathcal{T} is already captured by Theorem 5.42. Hence, the remaining cases are those that involve a TBox which is not FO-rewritable. Ideally, one would like to have a full classification of all pairs (q, \mathcal{T}) according to whether or not an FPRAS exists. Since the above results indicate that pABoxes with event expressions in DNF are promising we take them into account as well.

Again, if a query q is FO-rewritable relative to \mathcal{T} , then there is an FPRAS both for ipOBDA(q, \mathcal{T}) and pOBDA(q, \mathcal{T}) restricted to pABoxes with event expressions in DNF (by Theorem 5.42). If, on the other hand, q is not FO-rewritable relative to \mathcal{T} , the picture gets less positive. In what follows, we illustrate the computational behavior for a single pair of query and TBox and argue that (some of) our observations can be lifted to all non-FO rewritable pairs using Lemma 5.33.

Let us introduce a reliability problem for graphs. A *reliability network* is a tuple $G = (V, E, p)$ where (V, E) is a directed graph and p is a mapping that associates to every edge $e \in E$ a probability $p(e)$ that the edge is present. Intuitively, edges fail independently with probability $1 - p(e)$. The semantics is defined as for pABoxes, that is, each subset $E' \subseteq E$ is a possible world with probability $p(E') = \prod_{e \in E'} p(e) \cdot \prod_{e \in E \setminus E'} (1 - p(e))$. Given two vertices $s, t \in V$, we define the probability $p_{s,t}(G)$ that s, t are connected as the sum of the probability of all possible worlds such that there is a path from s to t .

s - t -RELIABILITY

INPUT: reliability network $G = (V, E, p)$, two vertices $s, t \in V$

OUTPUT: $p_{s,t}(G)$.

It is well-known that this problem is #P-hard [126], but it is open whether the problem can be approximated; an FPRAS is known to exist only under strong assumptions on G [132]. Note, however, the existence of an FPRAS for the ‘dual’ problem: given a reliability network, what is the probability that (V, E) becomes disconnected [87].

Theorem 5.44. *For $q = A(x)$ and $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$, $\text{ipOBDA}(q, \mathcal{T})$ and s - t -reliability are equivalent under polynomial time reductions, thus, the latter admits an FPRAS iff the former does.*

Proof. For the reduction from s - t -reliability to $\text{ipOBDA}(q, \mathcal{T})$, assume a reliability network $G = (V, E, p)$, define the ipABox \mathcal{A} by adding for all $(u, v) \in E$ the assertion $r(u, v)$ to \mathcal{A} and annotating it with probability $p(e)$. Add additionally the assertion $A(t)$. It is not difficult to verify that $p(\mathcal{T}, \mathcal{A} \models A(s)) = p_{s,t}(G)$. For the converse reduction, assume some ipABox \mathcal{A} . Construct a reliability network $G = (V, E, p)$ as:

- $V = \text{Ind}(\mathcal{A}) \cup \{x\}$;
- $E = \{(a, b) \mid r(a, b) \in \mathcal{A}\} \cup \{(a, x) \mid A(a) \in \mathcal{A}\}$;
- for an edge $e = (a, b) \in E$, define $p(e)$ as $p(r(a, b))$;
- for an edge $e = (a, x) \in E$, define $p(e) = 1$.

Also in this case, it is not hard to verify that $p(\mathcal{T}, \mathcal{A} \models A(a)) = p_{a,x}(G)$ for all $a \in \text{Ind}(\mathcal{A})$. ■

We leave a generalization of Point (1) as interesting future work. However, let us note that, when allowing pABoxes with DNF annotations, no FPRASes exist anymore if the query is not FO-rewritable relative to the TBox. In particular, we obtain the following characterization of FPRAS existence.

Theorem 5.45. *For Boolean connected CQs q and \mathcal{ELI} -TBoxes \mathcal{T} , there is an FPRAS for $\text{pOBDA}(q, \mathcal{T})$ over pABoxes restricted to DNF event expressions iff q is FO-rewritable relative to \mathcal{T} .*

Proof. The “if”-direction is a consequence of Theorem 5.42. For the “only if”-direction assume that q is not FO-rewritable relative to \mathcal{T} . We reduce from counting #SAT which has no FPRAS unless $\text{NP} = \text{PTime}$. Let φ be a CNF formula with k clauses $\varphi_1, \dots, \varphi_k$. By Lemma 5.33 there is an ABox $\mathcal{A} = \widehat{\mathcal{A}}_k$ containing a chain

$$R_1(a_1, a_2), \dots, R_k(a_k, a_{k+1})$$

such that $\mathcal{T}, \mathcal{A} \models q$ and removing one of $R_i(a_i, a_{i+1})$ leads to the fact that q is not implied anymore. Define the pABox (\mathcal{A}, e, p) as follows:

- $e(R_i(a_i, a_{i+1})) = \varphi_i$ for each $1 \leq i \leq k$;
- $e(\alpha) = x \vee \neg x$ for all assertions $\alpha \in \mathcal{A} \setminus \{R_1(a_1, a_2), \dots, R_k(a_k, a_{k+1})\}$;
- $p(x) = 1/2$ for all variables x occurring in φ .

We then have that every truth assignment v to variables in φ corresponds to a world $\mathcal{A}_v \subseteq \mathcal{A}$, and vice versa. Moreover, we have that $\mathcal{T}, \mathcal{A}_v \models A(a_0)$ iff $v \models \varphi$; hence $p(\mathcal{T}, \mathcal{A} \models A(a_0)) = \#\varphi/2^n$ where n is the number of variables in φ . ■

5.7 Conclusion and Future Directions

We have introduced the framework for ontology-based access to probabilistic data and we have analyzed the data complexity of computing answer probabilities therein. We believe that the introduced setup is of general interest and potentially useful for a wide range of applications including the already mentioned information extraction, machine translation, and dealing with data that arises from sensor networks. All these applications can potentially benefit from a fruitful interplay between ontologies and probabilities; in particular, we have argued that the ontology can help to reduce the uncertainty of the data. There are various directions for future work.

Generalizations of the presented results

The concrete dichotomy for DL-Lite covers a basic DL-Lite dialect and connected conjunctive queries. An obvious possibility for future work is to extend this result to more expressive versions of DL-Lite that, for example, allow for role hierarchy statements and functionality axioms. Although we believe that the techniques we presented are helpful for doing so, let us note that for some combinations the universal first-order rewritability is lost. This would require to investigate new techniques or to adapt those developed for \mathcal{ELI} . Another possibility would be to allow full conjunctive queries or even UCQs.

Another interesting direction is the generalization of the \mathcal{ELI} dichotomy to other logics. While the part when q is FO-rewritable relative to \mathcal{T} is always covered by Theorem 5.12,

recall that the proof for ‘non-FO-rewritability implies #P-hardness’ was non-trivial and required a good understanding of first-order rewritability in \mathcal{ELI} . There are two natural generalizations of \mathcal{ELI} : Horn description logics and \mathcal{ALCI} . Since the former are technically quite similar in traditional OBDA [17], we believe that an extension to them is not too hard. In contrast, the generalization towards \mathcal{ALCI} is probably more difficult, as FO-rewritability of conjunctive queries relative to \mathcal{ALCI} -TBoxes is not yet well-understood.

Implementation

From an application perspective, it would obviously be desirable to put our framework to work in an actual implementation and verify its utility in settings such as web data extraction or data cleaning. We believe there is legitimate hope for practicality given our positive results for computing approximate answer probabilities. Moreover, we have demonstrated that in many cases, mostly for DL-Lite, it can be implemented on top of existing probabilistic database systems, analogously to traditional OBDA. However, in an actual application, one is additionally interested in reasoning services different from standard query answering. A prominent such example is the computation of the *top-k answers*, that is, a potential user is not interested in *all* answer tuples and their probabilities but only in the k most probable ones, and sometimes not even in the probabilities then. Other helpful reasoning services include *explanation*, that is, explaining the high/low probability of query answers, and *support for feedback*, that is, allowing the user to correct wrongly extracted data. In this way, the system learns and provides increasingly better answers.

We have mostly dealt with query rewriting as a tool and ignored the potentially substantial increase of the query size involved. For addressing this issue in traditional OBDA, Kontchakov *et al.* introduced the ‘combined approach’ [94]. There, the query *and* the data is preprocessed—both in polynomial time—and the modified query is executed on the modified data; the TBox can be dropped. While obviously such an algorithm cannot exist for a #P-hard pair (q, \mathcal{T}) (unless PTIME = NP), this approach is potentially useful for the PTIME cases and thus of practical relevance.

Extensions of the Framework

Our framework allows for uncertain *data* only. However, similar settings might require also uncertainty in (different levels of) the TBox. For example, probabilities on the rules—corresponding to probabilities on concept inclusions—are allowed in [116, 50, 108]. While we believe that this is not too useful in the sketched application of information extraction, we can imagine that there are settings where such an extension is in fact necessary. It would be interesting to investigate the computational implications of such an extension. Note that an extension to ontologies as in Chapters 3 and 4 is semantically

delicate: a probabilistic ABox fixes a single distribution over (open) worlds whereas an ontology restricts possible distributions and it is not clear to which to give priority.

Orthogonally, one can imagine statements in the TBox that model statistical knowledge from the domain of discourse; this is done for example in [14, 92] in classical AI settings and in [34] in a probabilistic database setting. However, there are intricate semantical problems and it remains to be seen whether we can extend our framework to this direction.

6 Computational Complexity of the Product Logic $\mathbf{K} \times \mathbf{K}$

The purpose of this chapter is to study the complexity of the satisfiability problem in the product logic $\mathbf{K} \times \mathbf{K}$ and some variants thereof. The name ‘product logics’ stems from the fact that the semantics is defined based on frames obtained as direct products of standard frames. For example, the logic $\mathbf{K4} \times \mathbf{K}$ is interpreted on structures whose underlying frame is the asynchronous product of a transitive frame (for $\mathbf{K4}$) with an arbitrary frame (for \mathbf{K}).

Remarkably, there is a close semantical connection between product logics and ProbFO as introduced in Chapter 3, and thus its sublogic Prob \mathcal{EL} . To be more specific, ProbFO can be viewed as a generalization of the first-order modal logic $\mathbf{S5}_{\text{FO}}$.¹ For a formal definition of the logic $\mathbf{S5}_{\text{FO}}$, we refer the interested reader to the chapters about first-order modal logics in [52, Part III]. For our purposes, it suffices to mention that $\mathbf{S5}_{\text{FO}}$ has a ‘product like’ semantics and that one can view the domain of the $\mathbf{S5}$ -frame as the set of worlds. Recall that, in ProbFO, the worlds cannot be distinguished in the sense that when we compute $w(\varphi)$, we need to take *all* worlds into account; in a way, this is captured by the modal logic $\mathbf{S5}$ which is based on equivalence relations as frames. In fact, the mentioned connection becomes apparent in the *qualitative fragments*:

- ProbFO₀₁ from Section 3.1.3 is a notational variant of $\mathbf{S5}_{\text{FO}}$; and
- Prob \mathcal{EL} ₀₁ from Section 4.3.2 is a notational variant of the modal description logic $\mathbf{S5}_{\mathcal{EL}}$, an \mathcal{EL} -fragment of $\mathbf{S5}_{\text{FO}}$.

We refrain from giving the details of being a ‘notational variant’ and only note that, intuitively, one can replace a subformula $w(\varphi) > 0$ with $\Diamond\varphi$ and vice versa.

In general, it has been observed that the complexity (of satisfiability) in the product of two logics is often considerably more difficult than in the components. As an example, consider the basic modal logic \mathbf{K} and its variant $\mathbf{K4}$ for reasoning over transitive frames, for both of which satisfiability is PSPACE-complete [96]. In contrast, only nonelementary upper bounds were known for $\mathbf{K} \times \mathbf{K}$ and $\mathbf{K4} \times \mathbf{K}$ [54, 105]. Even worse, satisfiability becomes undecidable in $\mathbf{K} \times \mathbf{K} \times \mathbf{K}$ [74] and $\mathbf{K4} \times \mathbf{K4}$ [55].

The mentioned decidability result for $\mathbf{K} \times \mathbf{K}$ was first shown by Gabbay and Shehtman in 1998 [54] and using the same technique decidability in nonelementary time was

¹ $\mathbf{S5}$ is the modal logic based on frames where the accessibility relation is an equivalence.

shown for the closely related logics $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$ ($\mathbf{S4}$ ($\mathbf{S4}$ and $\mathbf{S5}_2$ are introduced in the chapter). In 2001, Marx and Mikuláš gave an alternative proof of the nonelementary upper bound for $\mathbf{K} \times \mathbf{K}$ and, additionally, proved NEXPTIME-hardness [105], which also carries over to the mentioned logics. Moreover, in the same paper they conjectured completeness for nonelementary time. Since then, the lower bound was not improved, and as a matter of fact, the precise complexity of all these logics is stated as interesting open question in [52]. Thus, the main aim of this chapter is to:

close the complexity gap for $\mathbf{K} \times \mathbf{K}$, $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$ by giving a nonelementary lower bound for the satisfiability problem.

Related Work

The study of many-dimensional modal logics was motivated by the good computational properties of modal logics and the need of combining different modalities in one applications, for example, when modeling the evolution of knowledge over time. The field received a lot of interest in the past twenty years; two excellent overviews are given by [52] and [20, Chapter 15]. Two important ways of combining two (or more) logics are *fusions* and *products*. Intuitively, the fusion of two logics corresponds to a *largely independent* combination of the logics. Due to that independence, fusions often lead to computationally well-behaved logics. For example the fusion of (unimodal) \mathbf{K} with itself is the bimodal logic \mathbf{K}_2 and thus of the same complexity, PSPACE. We refer the interested reader to the respective chapter in [52]. Products, on the other hand, have been shown to be a computationally delicate operation.

There are several product logics known to be complete for nonelementary time. Let us first mention $\mathbf{LTL} \times \mathbf{K}$, the product of linear temporal logic with \mathbf{K} , for which a nonelementary lower bound [52, 71] was proved using the ‘yardstick’ technique introduced by Stockmeyer [123]. Note that the proof of this theorem relies on the fact that \mathbf{LTL} includes the ‘until’-operator. This lower bound transfers via polynomial time reductions to, for example, $\mathbf{PDL} \times \mathbf{K}$ and $\mathbf{K}^C \times \mathbf{K}$ where \mathbf{K}^C is \mathbf{K} extended with the ‘common knowledge relation’ C —intuitively, C is the reflexive, transitive closure of the accessibility relation—see [52].

It is interesting to note that the asynchronous product (underlying the semantics of product logics) has served to model the behavior of concurrent processes in the realm of verification, see for instance [114]. There is, however, a slight difference in the adopted semantics. While in the field of many-dimensional logics, usually, a free interpretation of the propositional variables in the product structure is allowed, in verification, the interpretation is *inherited* from the underlying component structures. We will call the former ‘uninterpreted product structure’ and the latter ‘interpreted product structure’. Moreover, the verification community is typically interested in model checking instead of satisfiability.

Finally, let us mention that the trees we define to show the lower bound are similar to trees introduced in [48, Chapter 10]. We discuss their relationship in more detail later, when we have all our definitions available.

Contribution and Structure of the Chapter

The chapter is structured as follows. In Section 6.1, we will give the necessary preliminaries for products of modal logics. Most importantly, besides the standard ‘uninterpreted’ semantics used in the context of modal logics, we additionally introduce the mentioned ‘interpreted’ semantics from the field of verification. We first show that satisfiability in the uninterpreted semantics is at least as hard as in the interpreted semantics. This enables us to use the interpreted semantics throughout this chapter. In Section 6.2, we prove the nonelementary lower bound proceeding in three steps. First, we describe and introduce a family of nonelementarily branching trees, which are later used to encode numbers. In the second step, we give formulas which enforce these trees. Finally, we show that satisfiability of formulas with *switching depth* (the minimal modal depth among the two dimensions) ℓ is ℓ -NEXPTIME-hard for every $\ell \geq 1$. This confirms Marx and Mikuláš conjecture and implies the main result of this chapter: satisfiability in $\mathbf{K} \times \mathbf{K}$ is hard for nonelementary time. In Section 6.3, we use the obtained result to derive nonelementarily lower bounds for the variants $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$. In Section 6.4, we conclude the chapter, give some related open problems, and point out an additional application of our lower bound technique.

6.1 Preliminaries

First, we introduce the syntax and semantics of the two-dimensional modal logic $\mathbf{K} \times \mathbf{K}$ along the lines of the standard text book by Gabbay *et al.* [52]. Then, we formally define the problems studied in this chapter. Finally, we state and prove some important facts about bisimulation equivalence for product structures.

6.1.1 Many-dimensional modal logics

Let us fix a countable set of *accessibility relations* \mathbb{A} and a countable set of *propositional variables* \mathbb{P} . *Formulas of multimodal logic* are defined by the following grammar, where a and p range over \mathbb{A} and \mathbb{P} , respectively:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond_a \varphi.$$

We introduce the usual abbreviations $\top = p \vee \neg p$ for some $p \in \mathbb{P}$, $\perp = \neg\top$, $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, and $\Box_a \varphi = \neg\Diamond_a \neg\varphi$. For $\mathbf{A} \subseteq \mathbb{A}$ and $\mathbf{P} \subseteq \mathbb{P}$, we say that (\mathbf{A}, \mathbf{P}) is a *signature* and φ is an (\mathbf{A}, \mathbf{P}) -formula if the set of accessibility relations (respectively, the

set of propositional variables) that appear in φ is a subset of \mathbf{A} (respectively, \mathbf{P}). With $|\varphi|$ we denote the length of φ , that is, the number of symbols required to write φ down.

Multimodal logic formulas are interpreted in *pointed Kripke structures*, or just *pointed structures*. Given finite sets of accessibility relations $\mathbf{A} \subseteq \mathbf{A}$ and propositions $\mathbf{P} \subseteq \mathbf{P}$, an (\mathbf{A}, \mathbf{P}) -Kripke structure \mathfrak{S} consists of

- a relational structure or *Kripke frame* $\mathfrak{F} = (W, \{\xrightarrow{a} \mid a \in \mathbf{A}\})$ where W is a nonempty set of *worlds* and each \xrightarrow{a} is a binary *accessibility relation* over W , and
- a family of *interpretations* $\{W_p \subseteq W \mid p \in \mathbf{P}\}$.

Given a Kripke structure \mathfrak{S} , we refer to the frame underlying \mathfrak{S} with $\mathfrak{F}(\mathfrak{S})$. Moreover, given a frame $(W, \{\xrightarrow{a} \mid a \in \mathbf{A}\})$ we write $s \xrightarrow{a} t$ instead of $(s, t) \in \xrightarrow{a}$ and call each such t a *successor of s* . A *pointed (\mathbf{A}, \mathbf{P}) -structure* is a pair (\mathfrak{S}, s) where \mathfrak{S} is an (\mathbf{A}, \mathbf{P}) -structure and s is a world of \mathfrak{S} .

The *truth relation* $(\mathfrak{S}, s) \models \varphi$ for (\mathbf{A}, \mathbf{P}) -formulas φ and pointed (\mathbf{A}, \mathbf{P}) -structures (\mathfrak{S}, s) is given by structural induction on the definition of φ :

$$\begin{aligned} (\mathfrak{S}, s) &\models p && \text{iff } s \in W_p \\ (\mathfrak{S}, s) &\models \neg\varphi && \text{iff not } (\mathfrak{S}, s) \models \varphi \\ (\mathfrak{S}, s) &\models \varphi_1 \wedge \varphi_2 && \text{iff } (\mathfrak{S}, s) \models \varphi_1 \text{ and } (\mathfrak{S}, s) \models \varphi_2 \\ (\mathfrak{S}, s) &\models \Diamond_a \varphi && \text{iff } \exists s' : s \xrightarrow{a} s' \text{ and } (\mathfrak{S}, s') \models \varphi. \end{aligned}$$

A pointed (\mathbf{A}, \mathbf{P}) -structure (\mathfrak{S}, s) *satisfies* an (\mathbf{A}, \mathbf{P}) -formula φ if $(\mathfrak{S}, s) \models \varphi$. In this case, we say that (\mathfrak{S}, s) is a *model* of φ . Moreover, we call φ *satisfiable* if it has a model.

A frame $\mathfrak{F} = (W, \{\xrightarrow{a} \mid a \in \mathbf{A}\})$ is a *tree* if

- $W \subseteq A^*$ is a prefix-closed set of words;
- \xrightarrow{a} and \xrightarrow{b} are disjoint for $a \neq b$;
- for all $s, t \in W$ and $a \in \mathbf{A}$ we have $s \xrightarrow{a} t$ if, and only if $t = sa$.

We will refer with *root* to the node $\varepsilon \in W$. We also call a structure \mathfrak{S} a *tree* if its underlying frame $\mathfrak{F}(\mathfrak{S})$ is a tree.

Fix nonempty, finite, and disjoint sets $\mathbf{A}_1, \mathbf{A}_2 \subseteq \mathbf{A}$ of accessibility relations and nonempty, finite, and disjoint sets $\mathbf{P}_1, \mathbf{P}_2 \subseteq \mathbf{P}$ of propositional variables. Let $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$ and $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2$. For frames $\mathfrak{F}_i = (W_i, \{\xrightarrow{a_i} \mid a_i \in \mathbf{A}_i\})$, $i \in \{1, 2\}$, we define the *asynchronous product* $\mathfrak{F}_1 \times \mathfrak{F}_2$ as the frame $(W, \{\xrightarrow{a} \mid a \in \mathbf{A}\})$ where

- $W = W_1 \times W_2$, and
- for each $\bar{s} = \langle s_1, s_2 \rangle \in W$ and $\bar{t} = \langle t_1, t_2 \rangle \in W$ we have $\bar{s} \xrightarrow{a} \bar{t}$ if and only if:
 - $a \in \mathbf{A}_1$ implies $s_1 \xrightarrow{a_1} t_1$ and $s_2 = t_2$; and

– $a \in A_2$ implies $s_2 \xrightarrow{a}_2 t_2$ and $s_1 = t_1$.

We define two kinds of product structures, namely *uninterpreted product structures* and *interpreted product structures*. An (A, P) -structure $\mathfrak{S} = (W, \{\xrightarrow{a} \mid a \in A\}, \{W_p \mid p \in P\})$ is

- an *uninterpreted product structure* if $\mathfrak{F}(\mathfrak{S}) = \mathfrak{F}_1 \times \mathfrak{F}_2$, where each \mathfrak{F}_i is a frame $(W_i, \{\xrightarrow{a}_i \mid a \in A_i\})$.
- an *interpreted product structure* if $\mathfrak{F}(\mathfrak{S}) = \mathfrak{F}(\mathfrak{S}_1) \times \mathfrak{F}(\mathfrak{S}_2)$ for two structures $\mathfrak{S}_i = (W_i, \{\xrightarrow{a}_i \mid a \in A_i\}, \{W_{p,i} \mid p \in P_i\})$ and for all $i \in \{1, 2\}$, $p \in P_i$ we have:

$$\langle s_1, s_2 \rangle \in W_p \text{ if and only if } s_i \in W_{p,i}.$$

Note that in the former, there are no restrictions on how propositional variables are interpreted, while in the latter, the interpretation of the propositional variables in \mathfrak{S} is inherited from the component structures \mathfrak{S}_1 and \mathfrak{S}_2 . We also write $\mathfrak{S} = \mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2$, where the superscript “id” indicates that the interpretation of propositional variables is *identical* to the respective component structure. To the best of our knowledge, interpreted product structures have not been considered in the field of many-dimensional modal logics; however, they will turn out more convenient for enforcing structures, as we have more control of the propositions. Note that the interpreted product structures introduced in [114] are more general in that they are parametrized by *how* the interpretations are inherited. However, for our purposes this simplified definition suffices. In case of *pointed* structures (\mathfrak{S}_1, s_1) , (\mathfrak{S}_2, s_2) , the interpreted product structure is $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \langle s_1, s_2 \rangle)$.

Example 6.1. As an example, consider Figure 6.1. In parts (a) and (b), there are two structures \mathfrak{S}_1 and \mathfrak{S}_2 based on disjoint (A_1, P_1) and (A_2, P_2) with $P_1 = \{p\}$, $P_2 = \{q\}$, $A_1 = \{\downarrow\}$, and $A_2 = \{\rightarrow\}$. In part (c), there is the interpreted product of \mathfrak{S}_1 and \mathfrak{S}_2 . Note that when making a \rightarrow -transition, i.e. a transition in A_2 , the interpretation of p , a proposition in P_1 , does not change. In particular, we have for all i, j that

$$(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \langle i, j \rangle) \models (p \rightarrow \Box_{\rightarrow} p) \wedge (\Diamond_{\rightarrow} p \rightarrow p).$$

This is the crucial property that distinguishes interpreted from uninterpreted product structures, like the one in part (d) of Figure 6.1 which is based on the frame $\mathfrak{F}(\mathfrak{S}_1) \times \mathfrak{F}(\mathfrak{S}_2)$. There, the above formula is not satisfied in every $\langle i, j \rangle$.

We will also need the notion of *switching depth*, introduced in [105]. Given (A, P) as above, i.e., $A = A_1 \uplus A_2$ and $P = P_1 \uplus P_2$, and an (A, P) -formula φ , we define $\text{depth}_i(\varphi)$

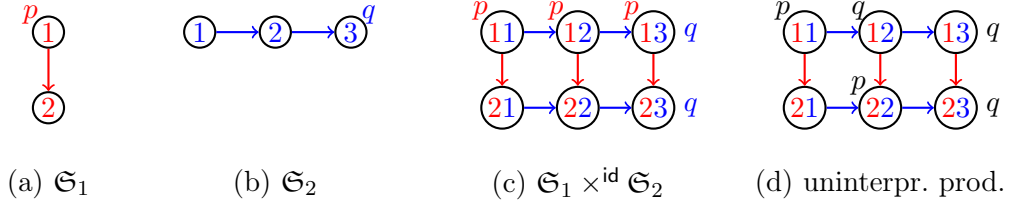


Figure 6.1: Example of some product structures.

for $i \in \{1, 2\}$ inductively by:

$$\begin{aligned}
 \text{depth}_i(p) &= 0 & p \in \mathbf{P} \\
 \text{depth}_i(\neg\varphi) &= \text{depth}_i(\varphi) \\
 \text{depth}_i(\varphi_1 \wedge \varphi_2) &= \max\{\text{depth}_i(\varphi_1), \text{depth}_i(\varphi_2)\} \\
 \text{depth}_i(\Diamond_a \varphi) &= \text{depth}_i(\varphi) & \text{if } a \notin A_i \\
 \text{depth}_i(\Diamond_a \varphi) &= \text{depth}_i(\varphi) + 1 & \text{if } a \in A_i
 \end{aligned}$$

and the switching depth of φ as $\min\{\text{depth}_1(\varphi), \text{depth}_2(\varphi)\}$. As an example, let $A_1 = \{a\}$, $A_2 = \{b\}$, and $p, q \in \mathbf{P}$. Then the formulas $p \wedge q$ and $\Diamond_a \Diamond_a p$ have switching depth 0; the formulas $\Diamond_a \Diamond_b(p \wedge q)$ and $\Diamond_b(p \vee \Box_a q)$ have switching depth 1; finally, the formula $\Diamond_a \Diamond_b \Diamond_a \Diamond_b \top$ has switching depth 2.

6.1.2 Decision problems

Let us say that φ is *(un-)interpreted satisfiable* if there is an (un-)interpreted pointed product structure (\mathfrak{S}, s) satisfying φ . Throughout the chapter, we will focus on the following decision problems.

\mathbf{K}^2 -SAT

INPUT: (A, \mathbf{P}) -formula φ

OUTPUT: Is φ uninterpreted satisfiable?

\mathbf{K}_{id}^2 -SAT

INPUT: (A, \mathbf{P}) -formula φ

OUTPUT: Is φ interpreted satisfiable?

Note that, here and in what follows, the partition of A and \mathbf{P} into $A_1 \uplus A_2$ and $\mathbf{P}_1 \uplus \mathbf{P}_2$, respectively, is rather implicit, but it will be always understood from the context. In particular, all results already hold for one accessibility relation for each dimension.

To the best of our knowledge, \mathbf{K}_{id}^2 -SAT was not studied so far, since the interpreted semantics stems from the verification community who typically studies model checking. However, considering *interpreted* product structures is more convenient as, intuitively,

we have more control of the propositions. Indeed, it turns out that for showing lower bounds, we can restrict our attention to \mathbf{K}_{id}^2 -SAT: the following proposition shows that \mathbf{K}^2 -SAT is at least as difficult as \mathbf{K}_{id}^2 -SAT.

Proposition 6.2. *There is a polynomial time many-one reduction from \mathbf{K}_{id}^2 -SAT to \mathbf{K}^2 -SAT which preserves the switching depth.*

Proof. Let $\mathbf{A} = \mathbf{A}_1 \uplus \mathbf{A}_2$ be the set of accessibility relations, $\mathbf{P} = \mathbf{P}_1 \uplus \mathbf{P}_2$ be the set of propositional variables, and φ be some (\mathbf{A}, \mathbf{P}) -formula. The idea is to give a formula χ that admits only product models which are in fact *interpreted* product structures, in particular, φ is interpreted satisfiable if and only if $\varphi \wedge \chi$ is uninterpreted satisfiable.

We need the following definition. The set of *modal sequences* $\text{ms}(\psi) \subseteq \mathbf{A}^*$ of a formula ψ is inductively defined as follows:

$$\begin{aligned} \text{ms}(p) &= \{\varepsilon\} \\ \text{ms}(\neg\psi) &= \text{ms}(\psi) \\ \text{ms}(\psi_1 \wedge \psi_2) &= \text{ms}(\psi_1) \cup \text{ms}(\psi_2) \\ \text{ms}(\Diamond_a \psi) &= (\{a\} \cdot \text{ms}(\psi)) \cup \{\varepsilon\} \end{aligned}$$

Intuitively, the set $\text{ms}(\psi)$ of modal sequences denotes the set of all sequences of modalities that can be found along the path to some node in the syntax tree of ψ . We note that $|\text{ms}(\varphi)| \leq |\varphi|$ and that the maximal length of an element of $\text{ms}(\varphi)$ is at most $|\varphi|$.

If $w = a_1 \cdots a_n \in \mathbf{A}^*$ we denote with \Box_w the sequence of boxes $\Box_{a_1} \cdots \Box_{a_n}$. Particularly, \Box_ε is the empty sequence of boxes. Moreover, define the relation $\xrightarrow{w} = \xrightarrow{a_1} \circ \xrightarrow{a_2} \circ \cdots \circ \xrightarrow{a_n}$. For $i \in \{1, 2\}$ and a word $w \in \mathbf{A}^*$ let $w \setminus i \in (\mathbf{A} \setminus \mathbf{A}_i)^*$ be the word that results from w by removing all occurrences of all symbols from \mathbf{A}_i and let $w \upharpoonright i \in \mathbf{A}_i^*$ be the word that results from w by removing all occurrences of all symbols from $\mathbf{A} \setminus \mathbf{A}_i$. We define the following formula χ :

$$\chi = \bigwedge_{i \in \{1, 2\}} \bigwedge_{w \in \text{ms}(\varphi)} \bigwedge_{p \in \mathbf{P}_i} \Box_{w \upharpoonright i} ((p \rightarrow \Box_{w \setminus i} p) \wedge (\Diamond_{w \setminus i} p \rightarrow p))$$

We define $\varphi' = \varphi \wedge \chi$. Note that φ' has the same switching depth as φ and can be constructed in polynomial time from φ . Therefore it suffices to show that φ is interpreted satisfiable if and only if φ' is uninterpreted satisfiable.

Since every interpreted product satisfies χ , it follows that φ' is uninterpreted satisfiable if φ is interpreted satisfiable. For the other direction let $\mathfrak{S} = (W, \{\xrightarrow{a} \mid a \in \mathbf{A}\}, \{W_p \mid p \in \mathbf{P}\})$ be an (\mathbf{A}, \mathbf{P}) -structure such that $\mathfrak{F}(\mathfrak{S}) = \mathfrak{F}_1 \times \mathfrak{F}_2$ with $\mathfrak{F}_i = (W_i, \{\xrightarrow{a} \mid a \in \mathbf{A}_i\})$ and assume that $(\mathfrak{S}, \bar{s}_0) \models \varphi \wedge \chi$ for some $\bar{s}_0 \in W$. Let

$$W_R = \{\bar{s} \in W \mid \exists w \in \text{ms}(\varphi) : \bar{s}_0 \xrightarrow{w} \bar{s}\}.$$

We define for each $i \in \{1, 2\}$ an $(\mathbf{A}_i, \mathbf{P}_i)$ -structure $\mathfrak{S}_i = (W_i, \{\xrightarrow{a} \mid a \in A_i\}, \{V_p \mid p \in P_i\})$ with underlying frame $\mathfrak{F}(\mathfrak{S}_i) = \mathfrak{F}_i$ such that $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}_0) \models \varphi$. For giving the interpretations V_p we need the following statement, where we denote the i -th component of a tuple \bar{t} with $\bar{t}(i)$.

Claim 1. For all $\bar{s}, \bar{t} \in W_R$, $i \in \{1, 2\}$, and $p \in P_i$: if $\bar{s}(i) = \bar{t}(i)$ then $(\bar{s} \in W_p \Leftrightarrow \bar{t} \in W_p)$.

Proof of Claim 1. Since $\bar{s}, \bar{t} \in W_R$ there exist $w, v \in \text{ms}(\varphi)$ such that

$$\bar{s}_0 \xrightarrow{w} \bar{s} \quad \text{and} \quad \bar{s}_0 \xrightarrow{v} \bar{t}.$$

Since \mathfrak{S} is a product model and $\bar{s}(i) = \bar{t}(i)$, there exists some \bar{r} with

$$\bar{s}_0 \xrightarrow{w \setminus i} \bar{r}, \quad \bar{s}_0 \xrightarrow{v \setminus i} \bar{r}, \quad \bar{r} \xrightarrow{w \setminus i} \bar{s}, \quad \bar{r} \xrightarrow{v \setminus i} \bar{t}.$$

Since $(\mathfrak{S}, \bar{s}_0) \models \chi$, we get $\bar{s} \in W_p \Rightarrow \bar{r} \in W_p \Rightarrow \bar{t} \in W_p$ and analogously $\bar{t} \in W_p \Rightarrow \bar{s} \in W_p$, which proves Claim 1.

Let us now define for all $p \in P_i$

$$V_p = \{\bar{s}(i) \in W_i \mid \bar{s} \in W_R \cap W_p\}.$$

With Claim 1, we get for all $\bar{s} \in W_R$:

$$\bar{s} \in W_p \Leftrightarrow \bar{s}(i) \in V_p \tag{6.1}$$

It remains to show that $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}_0) \models \varphi$. For a sequence $w \in \text{ms}(\varphi)$, let $\text{sub}_w(\varphi)$ be the set of all subformulas ψ of φ such that in the syntax tree for φ there exists a path to an occurrence of ψ such that w is the sequence of modalities along this path. We prove by induction on the structure of a subformula $\psi \in \text{sub}_w(\varphi)$ that for all $\bar{s} \in W_R$ with $\bar{s}_0 \xrightarrow{w} \bar{s}$:

$$(\mathfrak{S}, \bar{s}) \models \psi \Leftrightarrow (\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}) \models \psi.$$

For the induction base consider a propositional variable $p \in P$ and assume that $p \in P_i$ and $\bar{s} \in W_R$. We get:

$$(\mathfrak{S}, \bar{s}) \models p \Leftrightarrow \bar{s} \in W_p \stackrel{(6.1)}{\Leftrightarrow} \bar{s}(i) \in V_p \Leftrightarrow (\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}) \models p$$

For the induction step, the operators \wedge and \neg are straightforward. Finally, let $\psi = \Diamond_a \theta \in \text{sub}_w(\varphi)$ and assume that $\bar{s}_0 \xrightarrow{w} \bar{s}$. Hence, $\theta \in \text{sub}_{wa}(\varphi)$. We have:

$$\begin{aligned} (\mathfrak{S}, \bar{s}) \models \Diamond_a \theta &\Leftrightarrow \exists \bar{s}' : \bar{s} \xrightarrow{a} \bar{s}' \wedge (\mathfrak{S}, \bar{s}') \models \theta \\ &\stackrel{\text{hyp}}{\Leftrightarrow} \exists \bar{s}' : \bar{s} \xrightarrow{a} \bar{s}' \wedge (\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}') \models \theta \\ &\Leftrightarrow (\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}) \models \Diamond_a \theta \end{aligned}$$

Note that we can apply induction hypothesis since $w \in \text{ms}(\varphi)$ and $\Diamond_a \theta \in \text{sub}_w(\varphi)$ imply that $wa \in \text{ms}(\varphi)$ and, moreover, $\bar{s}_0 \xrightarrow{w} \bar{s}$ and $\bar{s} \xrightarrow{a} \bar{s}'$ imply $\bar{s}_0 \xrightarrow{wa} \bar{s}'$.

Since $\varphi \in \text{sub}_\varepsilon(\varphi)$, $\bar{s}_0 \in W_R$, and $\bar{s}_0 \xrightarrow{\varepsilon} \bar{s}_0$, this shows that $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \bar{s}_0) \models \varphi$. Overall, we have given a reduction from $\mathbf{K}_{\text{id}}^2\text{-SAT}$ to $\mathbf{K}^2\text{-SAT}$. ■

Note that the proof of this proposition does not make any assumption about the underlying frames. In particular, this implies that Proposition 6.2 holds also for restricted frame classes, for example, when we allow only for transitive frames in the components. We will need this later in Section 6.3.

6.1.3 Bisimulation equivalence

Let $\mathfrak{S} = (W, \{\xrightarrow{a} \mid a \in A\}, \{W_p \mid p \in P\})$ and $\mathfrak{S}' = (W', \{\xrightarrow{a} \mid a \in A\}, \{W'_p \mid p \in P\})$ be two (A, P) -structures. A *bisimulation between \mathfrak{S} and \mathfrak{S}'* is a binary relation $R \subseteq W \times W'$ such that for each $(s, s') \in R$ the following holds:

- (1) $s \in W_p$ if and only if $s' \in W'_p$ for all $p \in P$,
- (2) for each $s \xrightarrow{a} t$ there exists $s' \xrightarrow{a} t'$ such that $(t, t') \in R$, and
- (3) for each $s' \xrightarrow{a} t'$ there exists $s \xrightarrow{a} t$ such that $(t, t') \in R$.

In case there is a bisimulation R between \mathfrak{S} and \mathfrak{S}' with $(s, s') \in R$ we say that (\mathfrak{S}, s) is *bisimilar to* (\mathfrak{S}', s') and write $(\mathfrak{S}, s) \sim (\mathfrak{S}', s')$ or, in case \mathfrak{S} and \mathfrak{S}' are clear from the context, $s \sim s'$. We also say that (\mathfrak{S}, s) and (\mathfrak{S}', s') are *equivalent up to bisimulation*. It is well-known that modal logic cannot distinguish between bisimilar structures, i.e., if $(\mathfrak{S}, s) \sim (\mathfrak{S}', s')$ then $(\mathfrak{S}, s) \models \varphi$ if and only if $(\mathfrak{S}', s') \models \varphi$ for all (A, P) -formulas φ , see for instance [19]. The following proposition (which is straightforward to prove) lifts this statement to many-dimensional modal logics, that is, modal logic formulas cannot distinguish between interpreted product structures whose components are bisimilar.

Proposition 6.3. *Let $A = A_1 \uplus A_2$, $P = P_1 \uplus P_2$, and for each $i \in \{1, 2\}$ assume two pointed (A_i, P_i) -structures (\mathfrak{S}_i, s_i) and (\mathfrak{S}'_i, s'_i) with $(\mathfrak{S}_i, s_i) \sim (\mathfrak{S}'_i, s'_i)$. Then we have $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \langle s_1, s_2 \rangle) \sim (\mathfrak{S}'_1 \times^{\text{id}} \mathfrak{S}'_2, \langle s'_1, s'_2 \rangle)$. In particular, $(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \langle s_1, s_2 \rangle)$ and $(\mathfrak{S}'_1 \times^{\text{id}} \mathfrak{S}'_2, \langle s'_1, s'_2 \rangle)$ satisfy the same (A, P) -formulas.*

Proof. Assume for $i \in \{1, 2\}$ structures $\mathfrak{S}_i = (W_i, \{\xrightarrow{a}_i \mid a \in A_i\}, \{W_{p,i} \mid p \in P_i\})$ and $\mathfrak{S}'_i = (W'_i, \{\xrightarrow{a}_i \mid a \in A_i\}, \{W'_{p,i} \mid p \in P_i\})$ and worlds $s_i \in W_i, s'_i \in W'_i$. Moreover, let $\mathfrak{S} = (W, \{\xrightarrow{a} \mid a \in A\}, \{W_p \mid p \in P\})$ and $\mathfrak{S}' = (W', \{\xrightarrow{a} \mid a \in A\}, \{W'_p \mid p \in P\})$ be the respective interpreted products $\mathfrak{S} = \mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2$ and $\mathfrak{S}' = \mathfrak{S}'_1 \times^{\text{id}} \mathfrak{S}'_2$ and assume for each $i \in \{1, 2\}$ a bisimulation $R_i \subseteq W_i \times W'_i$ such that $(s_i, s'_i) \in R_i$. We claim that

$$R = \{(\langle t_1, t_2 \rangle, \langle t'_1, t'_2 \rangle) \mid \text{for all } i \in \{1, 2\} : (t_i, t'_i) \in R_i\}$$

is a bisimulation. For showing this, assume $(\bar{t}, \bar{t}') \in R$ where $\bar{t} = \langle t_1, t_2 \rangle \in W_1 \times W_2$ and $\bar{t}' = \langle t'_1, t'_2 \rangle \in W'_1 \times W'_2$. Hence, $t_i \sim t'_i$ and thus $t_i \in W_{p,i} \Leftrightarrow t'_i \in W'_{p,i}$ for each $p \in P_i$ and each $i \in \{1, 2\}$. By definition of interpreted product structures, we get $\bar{t} \in W_p$ if and only if $\bar{t}' \in W'_p$ for each $p \in P$. This establishes point (1) of R being a bisimulation. For proving point (2), let us assume $\bar{t} \xrightarrow{a} \bar{u}$, where $\bar{u} = \langle u_1, u_2 \rangle \in W_1 \times W_2$. Then there exists some $i \in \{1, 2\}$ such that $a \in A_i$, $t_i \xrightarrow{a} u_i$ and $t_{3-i} = u_{3-i}$. Since $(t_i, t'_i) \in R_i$ and $t_i \xrightarrow{a} u_i$ there exists some $u'_i \in W'_i$ such that $t'_i \xrightarrow{a'} u'_i$ and $(u_i, u'_i) \in R_i$. Set $\bar{u}' = \langle u'_1, u'_2 \rangle$ with u'_i as mentioned and $u'_{3-i} = t'_{3-i}$. Thus, we have $\bar{t}' \xrightarrow{a'} \bar{u}'$. Moreover, by construction we have $(u_j, u'_j) \in R_j$ for each $j \in \{1, 2\}$, hence $(\bar{u}, \bar{u}') \in R$ by definition of R . This establishes point (2). Point (3) can be proved analogously.

In particular, we have $(\mathfrak{S}, \langle s_1, s_2 \rangle) \sim (\mathfrak{S}', \langle s'_1, s'_2 \rangle)$. Thus, $(\mathfrak{S}, \langle s_1, s_2 \rangle)$ and $(\mathfrak{S}', \langle s'_1, s'_2 \rangle)$ satisfy the same (A, P) -formulas. \blacksquare

6.2 \mathbf{K}^2 -SAT is hard for nonelementary time

The goal of this section is to show a nonelementary lower bound for \mathbf{K}^2_{id} -SAT. We proceed in three steps:

- (1) define a family of trees and show how they encode elementary numbers;
- (2) give a family of formulas that enforce the trees previously defined;
- (3) use these formulas to show ℓ -NEXPTIME hardness for every $\ell \geq 1$.

To make (1) and (2) more precise, recall the function $\text{Tower} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, defined as

$$\begin{aligned} \text{Tower}(0, n) &= n && \text{for all } n \geq 0; \\ \text{Tower}(\ell + 1, n) &= 2^{\text{Tower}(\ell, n)} && \text{for all } \ell, n \geq 0. \end{aligned}$$

The goal is to give a family of formulas $\{\varphi_{\ell, n} \mid \ell, n \geq 0\}$ (over a signature to be specified later) such that for each $\ell, n \in \mathbb{N}$ the following holds:

- $|\varphi_{\ell, n}| \leq \exp(\ell) \cdot \text{poly}(\ell, n)$, and
- if $(\mathfrak{S} \times^{\text{id}} \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{\ell, n}$, then both (\mathfrak{S}, s) and $(\bar{\mathfrak{S}}, \bar{s})$ are *bisimilar* to particular tree structures and s and \bar{s} have at least $\text{Tower}(\ell, n)$ successors in \mathfrak{S} and $\bar{\mathfrak{S}}$, respectively. Moreover, these tree structures encode a number from $[0, \text{Tower}(\ell + 1, n) - 1]$.

6.2.1 Trees encoding numbers

Figure 6.2 illustrates the idea how a tree encodes a number: the node s has successors s_0, \dots, s_m such that each s_i is the root of some tree that encodes (inductively) the number i . Additionally, each node s_i might be labeled with a propositional variable b .

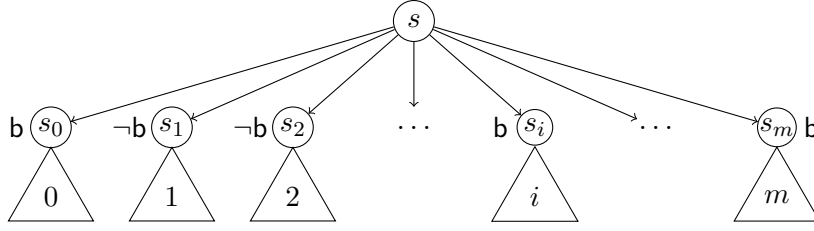


Figure 6.2: Intuition of trees encoding numbers.

The value encoded by s is the binary number (least significant bit to the left) that we can read off from the successors: bit i is 1 iff s_i satisfies \mathbf{b} . Hence, the number encoded by the tree in Figure 6.2 is of the form $100\dots 1$ and the i -th bit is 1. Note that the encoded number is in the interval $[0, 2^{m+1} - 1]$.

We need some additional propositional variables. In order to give a formal definition of the above described trees, fix a signature $(\{a\}, P_n)$ with

$$P_n = \{\mathbf{b}_0, \dots, \mathbf{b}_{n-1}, \mathbf{b}, \min_{\mathbf{b}}, \min_{\mathbf{b}}^{\leftarrow}, \min_{\neg \mathbf{b}}, \min_{\neg \mathbf{b}}^{\leftarrow}\}.$$

The following definition gives a family of tree structures over this signature. Note that these are implicitly *pointed* structures, pointed at their root. A tree $\Upsilon_{\ell,n}(j, V)$ has four parameters: ℓ, n determine the outdegree of $\text{Tower}(\ell, n)$, j is the encoded value, and V is the set of propositions from $P_{\text{aux}} = \{\mathbf{b}, \min_{\mathbf{b}}, \min_{\mathbf{b}}^{\leftarrow}, \min_{\neg \mathbf{b}}, \min_{\neg \mathbf{b}}^{\leftarrow}\}$ satisfied in the root of the tree. The definition is by induction on ℓ .

Definition 6.4. For $\ell = 0$, let $j \in [0, 2^n - 1]$ and $V \subseteq P_{\text{aux}}$. Then $\Upsilon_{0,n}(j, V)$ is a pointed $(\{a\}, P_n)$ -structure (\mathfrak{S}, s) with $\mathfrak{S} = (\{s\}, \emptyset, \{W_p \mid p \in P_n\})$ such that

- $j = \sum \{2^i \mid i \in [0, n-1], s \in W_{\mathbf{b}_i}\}$ and
- for each $p \in P_{\text{aux}}$ we have $(\mathfrak{S}, s) \models p$ iff $p \in V$.

For $\ell > 0$, let $j \in [0, \text{Tower}(\ell+1, n) - 1]$ and $V \subseteq P_{\text{aux}}$. We define the tree $\Upsilon_{\ell,n}(j, V)$ as follows. Set $m = \text{Tower}(\ell, n) - 1$ and let $I^+ \subseteq [0, m]$ be the unique set such that $j = \sum_{i \in I^+} 2^i$ and let $I^- = [0, m] \setminus I^+$. Now, fix a sequence $(\mathfrak{S}_0, s_0), \dots, (\mathfrak{S}_m, s_m)$ of trees of the form $\Upsilon_{\ell-1,n}(i, V_i)$, that is each (\mathfrak{S}_i, s_i) encodes the value i , such that for all $i \in [0, m]$ we have:

- (i) $(\mathfrak{S}_i, s_i) \models \mathbf{b}$ iff $i \in I^+$,
- (ii) $(\mathfrak{S}_i, s_i) \models \min_{\mathbf{b}}$ iff $i = \min(I^+)$,
- (iii) $(\mathfrak{S}_i, s_i) \models \min_{\neg \mathbf{b}}$ iff $i = \min(I^-)$,

(iv) $(\mathfrak{S}_i, s_i) \models \min_{\mathbf{b}}^{\leftarrow}$ iff $i < \min(I^+)$ or $I^+ = \emptyset$,

(v) $(\mathfrak{S}_i, s_i) \models \min_{\neg \mathbf{b}}^{\leftarrow}$ iff $i < \min(I^-)$ or $I^- = \emptyset$,

We obtain the tree $\Upsilon_{\ell,n}(j, V) = (\mathfrak{S}, s)$ by taking the union of all structures $\mathfrak{S}_0, \dots, \mathfrak{S}_m$, adding a world s , adding the pairs (s, s_i) to \xrightarrow{a} for all $i \in [0, m]$, and labeling s with precisely the propositions in V .

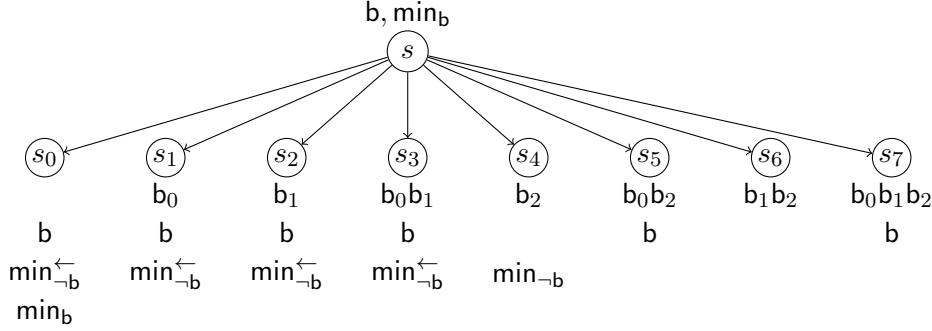
First note that this definition uniquely determines the trees $\Upsilon_{\ell,n}(j, V)$. For $\ell = 0$, this is easily seen; for $\ell > 0$ observe that j uniquely determines I^+ . By condition (i) the interpretation of \mathbf{b} in the worlds s_i is defined. Moreover, conditions (ii)-(v) imply (in this order) that:

- $\min_{\mathbf{b}}$ identifies the minimal i such that s_i satisfies \mathbf{b} ;
- $\min_{\neg \mathbf{b}}$ identifies the minimal i such that s_i satisfies $\neg \mathbf{b}$;
- $\min_{\mathbf{b}}^{\leftarrow}$ labels all successors left of $\min_{\mathbf{b}}$ (or all successors if $\min_{\mathbf{b}}$ does not exist);
- $\min_{\neg \mathbf{b}}^{\leftarrow}$ labels all successors left of $\min_{\neg \mathbf{b}}$ (or all successors if $\min_{\neg \mathbf{b}}$ does not exist).

As an example, Figure 6.3 shows the tree $\Upsilon_{1,3}(175, \{\mathbf{b}, \min_{\mathbf{b}}\})$. It is not hard to verify that each of the s_i is a tree $\Upsilon_{0,3}(i, V_i)$ for some V_i and that conditions (i)-(v) from Definition 6.4 are satisfied. In particular, the successors of s give rise to the binary number 11110101 (least significant bit to the left) which equals 175. As required, $\min_{\mathbf{b}}$ holds in the minimal position where \mathbf{b} holds and $\min_{\mathbf{b}}^{\leftarrow}$ holds in all positions left of $\min_{\mathbf{b}}$; and analogously for $\min_{\neg \mathbf{b}}$ and $\min_{\neg \mathbf{b}}^{\leftarrow}$.

We will sometimes drop V from the notation $\Upsilon_{\ell,n}(j, V)$ if we want to refer to the set of all trees of this form; in particular, we say ‘some $\Upsilon_{\ell,n}(j)$ ’ instead of ‘ $\Upsilon_{\ell,n}(j, V)$ for some set V ’ given that no confusion is possible. In the same way as $\Upsilon_{\ell,n}(j, V)$, we can define trees $\overline{\Upsilon}_{\ell,n}(j, V)$ over the signature $(\{\overline{a}\}, \overline{P}_n)$ where $\overline{P}_n = \{\overline{p} \mid p \in P_n\}$. In particular, $\overline{\Upsilon}_{\ell,n}(j, V)$ is obtained from $\Upsilon_{\ell,n}(j, V)$ by replacing \xrightarrow{a} by $\xrightarrow{\overline{a}}$ and every proposition $p \in P_n$ by $\overline{p} \in \overline{P}_n$.

It is worth mentioning that the defined trees $\Upsilon_{\ell,n}(j)$ are similar to the trees $\mathcal{T}(j)$ introduced in [48, Chapter 10] and used for example in [38]. In particular, they both represent the number j and have small depth, but high outdegree. However, there are some differences. Note first that the root of $\mathcal{T}(j)$ has a child for those numbers i such that the i -th bit in j is 1. In contrast, the root of $\Upsilon_{\ell,n}(j)$ has, independent of j , $\text{Tower}(\ell, n)$ children each corresponding to one bit position and the bits set to 1 are marked with the proposition \mathbf{b} . Moreover, as we use two-dimensional modal logic instead of first-order logic as in [48] to enforce our trees, we face two problems: First, we cannot enforce them up to isomorphism but only up to bisimulation equivalence. Second, as the logic is much weaker, we need some auxiliary propositional variables (or unary predicates). The particular difficulty is expressing a “less-than” or “successor” predicate and we use the propositions from P_{aux} for this purpose.


 Figure 6.3: The $(1, 3)$ -tree $\Upsilon_{1,3}(175, \{b, \min_b\})$.

6.2.2 Formulas enforcing the trees $\Upsilon_{\ell,n}(j)$

We are now ready to give the announced family of formulas $\{\varphi_{\ell,n} \mid \ell, n \geq 0\}$. Intuitively, the formulas $\varphi_{\ell,n}$ enforce *up to bisimulation equivalence* products of trees $\Upsilon_{\ell,n}(j)$ and $\bar{\Upsilon}_{\ell,n}(j)$; that is, any model $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle)$ of $\varphi_{\ell,n}$ has the property that there is some j and trees $\Upsilon_{\ell,n}(j), \bar{\Upsilon}_{\ell,n}(j)$ such that $(\mathfrak{S}, s) \sim \Upsilon_{\ell,n}(j)$ and $(\bar{\mathfrak{S}}, \bar{s}) \sim \bar{\Upsilon}_{\ell,n}(j)$.

The formulas we construct will be over the signature (A, P) with $A = \{a, \bar{a}\}$ and $P = P_n \cup \bar{P}_n$. We start with some auxiliary formulas $\text{eq}_{\ell,n}$, $\text{first}_{\ell,n}$, $\text{last}_{\ell,n}$, and $\text{succ}_{\ell,n}$ whose names indicate their meaning on the (interpreted) product of two trees $\Upsilon_{\ell,n}(j_1)$ and $\bar{\Upsilon}_{\ell,n}(j_2)$. For the sake of simplicity, we write $\Diamond, \Box, \bar{\Diamond}, \bar{\Box}$ instead of the modalities $\Diamond_a, \Box_a, \Diamond_{\bar{a}}, \Box_{\bar{a}}$, respectively.

For $\ell = 0$, the auxiliary formulas are defined as follows:

$$\begin{aligned} \text{eq}_{0,n} &= \bigwedge_{i \in [0, n-1]} b_i \leftrightarrow \bar{b}_i \\ \text{first}_{0,n} &= \bigwedge_{i \in [0, n-1]} \neg b_i \wedge \neg \bar{b}_i \\ \text{last}_{0,n} &= \bigwedge_{i \in [0, n-1]} b_i \wedge \bar{b}_i \\ \text{succ}_{0,n} &= \bigvee_{i \in [0, n-1]} (\neg b_i \wedge \bar{b}_i \wedge \bigwedge_{j \in [0, i-1]} (b_j \wedge \neg \bar{b}_j) \wedge \bigwedge_{j \in [i+1, n-1]} b_j \leftrightarrow \bar{b}_j) \end{aligned}$$

For $\ell > 0$ we define them as follows:

$$\begin{aligned} \text{eq}_{\ell,n} &= \Box \bar{\Box} (\text{eq}_{\ell-1,n} \rightarrow (b \leftrightarrow \bar{b})) \\ \text{first}_{\ell,n} &= \Box \neg b \wedge \bar{\Box} \neg \bar{b} \\ \text{last}_{\ell,n} &= \Box b \wedge \bar{\Box} \bar{b} \\ \text{succ}_{\ell,n} &= \Diamond \neg b \wedge \Box \bar{\Box} (\text{eq}_{\ell-1,n} \rightarrow ((\min_{-b} \leftrightarrow \bar{\min}_{-b}) \wedge ((\neg \min_{-b}^{\leftarrow} \wedge \neg \min_{-b}) \rightarrow (b \leftrightarrow \bar{b})))) \end{aligned}$$

The following lemma shows that these auxiliary formulas indeed express what they suggest to express. For brevity and since we only deal with the interpreted product we drop the superscript *id*.

Lemma 6.5. *Let $\ell, n \geq 0$, let $j_1, j_2 \in [0, \text{Tower}(\ell + 1, n) - 1]$, V_1, V_2 arbitrary and fix $\mathfrak{T} = \Upsilon_{\ell, n}(j_1, V_1)$ and $\overline{\mathfrak{T}} = \overline{\Upsilon}_{\ell, n}(j_2, V_2)$. Then the following holds:*

- (a) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{eq}_{\ell, n}$ if and only if $j_1 = j_2$.
- (b) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{first}_{\ell, n}$ if and only if $j_1 = j_2 = 0$.
- (c) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{last}_{\ell, n}$ if and only if $j_1 = j_2 = \text{Tower}(\ell + 1, n) - 1$.
- (d) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{succ}_{\ell, n}$ if and only if $j_2 = j_1 + 1$.

Proof. We show the statement by induction on ℓ . Let \mathfrak{T} and $\overline{\mathfrak{T}}$ be as in the lemma. Note first that the statement does not depend on V_1, V_2 , so we will drop them here. Moreover, let s and \overline{s} denote the roots of $\Upsilon_{\ell, n}(j_1)$ and $\overline{\Upsilon}_{\ell, n}(j_2)$, respectively.

For the induction base let $\ell = 0$. For (a) we have $(\Upsilon_{\ell, n}(j_1) \times \overline{\Upsilon}_{\ell, n}(j_2)) \models \text{eq}_{0, n}$ if and only if $(\mathbf{b}_i$ holds in $s \Leftrightarrow \overline{\mathbf{b}}_i$ holds in $\overline{s})$ for all $i \in [0, n - 1]$ if and only if $j_1 = j_2$. Both (b) and (c) can be proved in analogy to (a). For (d) we have $(\Upsilon_{\ell, n}(j_1) \times \overline{\Upsilon}_{\ell, n}(j_2)) \models \text{succ}_{0, n}$ if and only if there is some $i \in [0, n - 1]$ such that

- s does not satisfy \mathbf{b}_i and \overline{s} satisfies $\overline{\mathbf{b}}_i$,
- for each $j \in [0, i - 1]$: s satisfies \mathbf{b}_j and \overline{s} does not satisfy $\overline{\mathbf{b}}_j$, and
- for each $j \in [i + 1, n - 1]$: s satisfies \mathbf{b}_j if and only if \overline{s} satisfies $\overline{\mathbf{b}}_j$.

This is equivalent to $j_2 = j_1 + 1$.

For the induction step let $\ell > 0$. The cases (a), (b), and (c) are straightforward. Let us prove case (d). Recall that, according to Definition 6.4, there are two uniquely defined sets I_1^+ and I_2^+ representing (the bits set to 1 in the binary encoding of) j_1 and j_2 , respectively. The formula $\text{succ}_{\ell, n}$ states the following:

- there is a $k \in [0, \text{Tower}(\ell, n) - 1]$ such that $k \notin I_1^+$;
- if k_0 is the minimal $k \in [0, \text{Tower}(\ell, n) - 1]$ such that $k \notin I_1^+$, then k_0 is also the minimal $k \in [0, \text{Tower}(\ell, n) - 1]$ such that $k \in I_2^+$;
- for all $k_0 < k < \text{Tower}(\ell, n)$ we have $k \in I_1^+$ iff $k \in I_2^+$.

Thus, the binary representations (least significant bit to the left) of j_1 and j_2 are of the form $1^{k_0}0c_1 \dots c_m$ and $0^{k_0}1c_1 \dots c_m$ for some bits $c_1, \dots, c_m \in \{0, 1\}$; that is $j_2 = j_1 + 1$. ■

The next definition gives the family of (A, P) -formulas $\{\varphi_{\ell,n} \mid \ell, n \geq 0\}$; the subsequent theorem states that they satisfy the desired properties.

Definition 6.6. Set $\varphi_{0,n} = \text{eq}_{0,n} \wedge \Box \perp \wedge \Box \perp$ and for each $\ell \geq 1$ define $\varphi_{\ell,n}$, by induction on ℓ , as the conjunction of the following formulas:

- (1) $\bigwedge_{i \in [0, n-1]} \neg b_i \wedge \neg \bar{b}_i$
- (2) $\Box \Diamond \varphi_{\ell-1,n}$
- (3) $\Box \Diamond \varphi_{\ell-1,n}$
- (4) $\Diamond \Diamond (\varphi_{\ell-1,n} \wedge \text{first}_{\ell-1,n})$
- (5) $\Box (\Box \neg \text{last}_{\ell-1,n} \rightarrow \Diamond \text{succ}_{\ell-1,n})$
- (6) $\Box \Box (\text{eq}_{\ell-1,n} \rightarrow \bigwedge_{p \in P_n} (p \leftrightarrow \bar{p}))$
- (7) $\Diamond (\min_{\neg b} \vee \min_{\neg b}^{\leftarrow}) \wedge \Diamond (\min_b \vee \min_b^{\leftarrow})$
- (8) $\Box (((\min_{\neg b} \vee \min_b^{\leftarrow}) \rightarrow \neg b) \wedge ((\min_{\neg b}^{\leftarrow} \vee \min_b) \rightarrow b))$
- (9) $\Box \Box (\text{succ}_{\ell-1,n} \rightarrow \bigwedge_{x \in \{b, \neg b\}} ((\min_x \vee \min_x^{\leftarrow}) \leftrightarrow \min_x^{\leftarrow}))$

Theorem 6.7. For every $\ell, n \geq 0$ the following holds:

- (a) $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{\ell,n}$ if and only if there exists $j \in [0, \text{Tower}(\ell+1, n) - 1]$ such that (\mathfrak{S}, s) is bisimilar to some $\Upsilon_{\ell,n}(j)$ and $(\bar{\mathfrak{S}}, \bar{s})$ is bisimilar to some $\bar{\Upsilon}_{\ell,n}(j)$.
- (b) $|\varphi_{\ell,n}| \leq 3^\ell \cdot \text{poly}(\ell, n)$ and the formula $\varphi_{\ell,n}$ is computable in time $3^\ell \cdot \text{poly}(\ell, n)$.
- (c) The switching depth of $\varphi_{\ell,n}$ is ℓ .

Before giving the complete formal proof of Theorem 6.7, we want to give some intuition. Parts (b) and (c) are straightforward consequences of the definition of $\varphi_{\ell,n}$. For Part (a) observe that it is routine to verify that the product of any $\Upsilon_{\ell,n}(j)$ and $\bar{\Upsilon}_{\ell,n}(j)$ satisfies $\varphi_{\ell,n}$. The difficult part is to show that $\varphi_{\ell,n}$ enforces such models, that is, each model of $\varphi_{\ell,n}$ is of the form $\mathfrak{T} \times \bar{\mathfrak{T}}$, where \mathfrak{T} and $\bar{\mathfrak{T}}$ are bisimilar to structures $\Upsilon_{\ell,n}(j)$ and $\bar{\Upsilon}_{\ell,n}(j)$, respectively, for some j . Obviously, this is the case for $\varphi_{0,n}$.

For $\ell > 0$, let $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{\ell,n}$. By induction, formula (2) implies that for each successor t of s it holds that (\mathfrak{S}, t) is bisimilar to some $\Upsilon_{\ell-1,n}(i)$. Formula (3) implies the analogous property for every successor \bar{t} of \bar{s} .

Formulas (3)-(5) together imply that for every $i \in [0, \text{Tower}(\ell, n) - 1]$ there is a successor s_i of s such that (\mathfrak{S}, s_i) is bisimilar to some $\Upsilon_{\ell-1, n}(i)$ and, analogously, there is a successor \bar{s}_i of \bar{s} such that $(\bar{\mathfrak{S}}, \bar{s}_i)$ is bisimilar to some $\bar{\Upsilon}_{\ell-1, n}(i)$. By formula (4), there are such s_0 and \bar{s}_0 . The existence of s_0 and formula (5) imply the existence of \bar{s}_1 . Formula (3) implies that there is some s_1 , and so on.

Observe now that, in principle, there might be successors $s_i \neq s'_i$ of s such that (\mathfrak{S}, s_i) and (\mathfrak{S}, s'_i) are bisimilar to $\Upsilon_{\ell, n}(i, V)$ and $\Upsilon_{\ell, n}(i, V')$ for *different* sets $V \neq V'$. This is ruled out by applying formula (6) twice: For any proposition $p \in \mathbf{P}_n$ we have: p is satisfied in (\mathfrak{S}, s_i) if and only if \bar{p} is satisfied in $(\bar{\mathfrak{S}}, \bar{s}_i)$ if and only if p is satisfied in (\mathfrak{S}, s'_i) . Hence, we can talk about *the* successors s_i and \bar{s}_i , respectively.

The successors s_i and \bar{s}_i encode binary numbers N and \bar{N} , respectively, in the natural way: The i -th bit² of N is 1 if and only if (\mathfrak{S}, s_i) satisfies \mathbf{b} and analogously for \bar{N} . Note that formula (6) implies that $N = \bar{N}$.

Finally, formulas (7)-(9) ensure that the successors s_i and \bar{s}_i are labeled with the propositions $\min_{\mathbf{b}}, \min_{\neg \mathbf{b}}, \min_{\mathbf{b}}^{\leftarrow}, \min_{\neg \mathbf{b}}^{\leftarrow}$ and $\overline{\min_{\mathbf{b}}}, \overline{\min_{\neg \mathbf{b}}}, \overline{\min_{\mathbf{b}}^{\leftarrow}}, \overline{\min_{\neg \mathbf{b}}^{\leftarrow}}$, respectively, in a way such that (\mathfrak{S}, s) is bisimilar to some $\Upsilon_{\ell, n}(N)$ and $(\bar{\mathfrak{S}}, \bar{s})$ is bisimilar to some $\bar{\Upsilon}_{\ell, n}(N)$; that is, they ensure that conditions (ii)-(v) from Definition 6.4 are satisfied. This is actually the most subtle part of the following proof of Theorem 6.7.

Proof of Theorem 6.7. Part (c) is an immediate consequence of Definition 6.6.

We show part (b) by induction on ℓ starting with $\ell = 0$. For $\varphi_{0, n} = \bigwedge_{i \in [0, n-1]} \mathbf{b}_i \leftrightarrow \bar{\mathbf{b}}_i \wedge \Box \perp \wedge \bar{\Box} \perp$ the statement is trivial. Let now be $\ell > 0$. The formula $\varphi_{\ell-1, n}$ occurs 3 times in $\varphi_{\ell, n}$. The auxiliary formulas $\text{succ}_{\ell-1, n}$, $\text{eq}_{\ell-1, n}$, $\text{last}_{\ell-1, n}$, and $\text{first}_{\ell-1, n}$ are all polynomially sized in ℓ and n . Thus, overall we get $|\varphi_{\ell, n}| = 3 \cdot |\varphi_{\ell-1, n}| + \text{poly}(\ell, n)$. Thus, we obtain by induction hypothesis $|\varphi_{\ell, n}| = 3^\ell \cdot \text{poly}(\ell, n)$.

Let us finally prove part (a). With (1), (2), \dots , (9) we refer to the formulas from Definition 6.6.

“if”: We prove the “if”-direction by induction on ℓ . For the induction base, assume $\ell = 0$. Assume some $j \in [0, \text{Tower}(1, n) - 1] = [0, 2^n - 1]$ such that (\mathfrak{S}, s) is bisimilar to some $\Upsilon_{0, n}(j)$ and $(\bar{\mathfrak{S}}, \bar{s})$ is bisimilar to some $\bar{\Upsilon}_{0, n}(j)$. It is clear that $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \Box \perp \wedge \bar{\Box} \perp$. Moreover, Proposition 6.3 and Point (a) of Lemma 6.5 imply that $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \text{eq}_{0, n}$. Hence, $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{0, n}$.

For the induction step, assume $\ell \geq 1$. Let j be arbitrary in $[0, \text{Tower}(\ell + 1, n) - 1]$ and assume (\mathfrak{S}, s) is bisimilar to some $\Upsilon_{\ell, n}(j, V_1)$ and $(\bar{\mathfrak{S}}, \bar{s})$ is bisimilar to some $\bar{\Upsilon}_{\ell, n}(j, V_2)$. By Proposition 6.3, it suffices to show that $\Upsilon_{\ell, n}(j, V_1) \times \bar{\Upsilon}_{\ell, n}(j, V_2) \models \varphi_{\ell, n}$. Let r and \bar{r} be the root of $\Upsilon_{\ell, n}(j, V_1)$ and $\bar{\Upsilon}_{\ell, n}(j, V_2)$. Note that none of the formulas (1)-(9) refers to the propositions possibly in V_1, V_2 , so in what follows we drop V_1, V_2 .

²Again, the least significant bit is the 0-th bit.

Clearly, formula (1) holds in $\Upsilon_{\ell,n}(j) \times \bar{\Upsilon}_{\ell,n}(j)$ as neither $\Upsilon_{\ell,n}(j)$ satisfies any \mathbf{b}_i nor does $\bar{\Upsilon}_{\ell,n}(j)$ satisfy any proposition $\bar{\mathbf{b}}_i$.

For formula (2) let t be any successor of r . By Definition 6.4, the subtree \mathfrak{T} rooted in t is some $\Upsilon_{\ell-1,n}(i, V)$ for some $i \in [0, \text{Tower}(\ell, n) - 1]$ and a set V . Also by Definition 6.4, there is a successor \bar{t} of \bar{r} such that the subtree $\bar{\mathfrak{T}}$ rooted in \bar{t} is $\bar{\Upsilon}_{\ell-1,n}(i, V)$. By induction hypothesis, we have $\mathfrak{T} \times \bar{\mathfrak{T}} \models \varphi_{\ell-1,n}$. Formula (3) holds for analogous reasons.

For formula (4) observe that, by Definition 6.4, there are successors t and \bar{t} of r and \bar{r} , respectively, such that the subtrees \mathfrak{T} and $\bar{\mathfrak{T}}$ rooted in t and \bar{t} are $\Upsilon_{\ell-1,n}(0, V)$ and $\bar{\Upsilon}_{\ell-1,n}(0, \bar{V})$, respectively (for fixed sets V, \bar{V}). Point (b) of Lemma 6.5 implies $\mathfrak{T} \times \bar{\mathfrak{T}} \models \text{first}_{\ell-1,n}$. Hence, formula (4) is satisfied.

For formula (5) assume that t is an arbitrary successor of r . By Definition 6.4, the subtree \mathfrak{T} rooted in t is some $\Upsilon_{\ell-1,n}(i)$ for some $i \in [0, \text{Tower}(\ell, n) - 1]$. We distinguish the following cases on i .

- $i = \text{Tower}(\ell, n) - 1$. By Definition 6.4, there is a successor \bar{t} of \bar{r} such that the subtree $\bar{\mathfrak{T}}$ rooted in \bar{t} is some $\bar{\Upsilon}_{\ell-1,n}(i)$. By point (c) of Lemma 6.5, we have $\mathfrak{T} \times \bar{\mathfrak{T}} \models \text{last}_{\ell-1,n}$.
- $i < \text{Tower}(\ell, n) - 1$. By Definition 6.4, there is a successor \bar{t} of \bar{r} such that the subtree $\bar{\mathfrak{T}}$ rooted in \bar{t} is some $\bar{\Upsilon}_{\ell-1,n}(i+1)$. By point (d) of Lemma 6.5, we have $\mathfrak{T} \times \bar{\mathfrak{T}} \models \text{succ}_{\ell-1,n}$.

For formula (6) let t and \bar{t} be arbitrary successors of r and \bar{r} , respectively. There are $k, i \in [0, \text{Tower}(\ell, n) - 1]$ such that the subtree \mathfrak{T} (resp., $\bar{\mathfrak{T}}$) rooted in t (resp., \bar{t}) is some $\Upsilon_{\ell-1,n}(k)$ (resp., some $\bar{\Upsilon}_{\ell-1,n}(i)$). Now, assume that $\mathfrak{T} \times \bar{\mathfrak{T}} \models \text{eq}_{\ell-1,n}$. Point (a) of Lemma 6.5 implies that $k = i$. Recall that by conditions (i)-(v) of Definition 6.4 the interpretation of the propositions in t and \bar{t} is uniquely determined by $k = i$. Hence, a proposition $p \in P_n$ holds in t if and only if \bar{p} holds in \bar{t} .

For formula (7) observe that, by conditions (iii) and (v) in Definition 6.4, either there is some successor of r labeled with $\min_{\mathbf{b}}$ or all successors of r are labeled with $\min_{\mathbf{b}}^{\leftarrow}$. Thus, $\Upsilon_{\ell,n}(j) \times \bar{\Upsilon}_{\ell,n}(j) \models \Diamond(\min_{\mathbf{b}} \vee \min_{\mathbf{b}}^{\leftarrow})$. Similarly, conditions (ii) and (iv) imply that either there is some successor of t labeled with $\min_{\neg \mathbf{b}}$ or all successors of t are labeled with $\min_{\neg \mathbf{b}}^{\leftarrow}$. Hence, $\Upsilon_{\ell,n}(j) \times \bar{\Upsilon}_{\ell,n}(j) \models \Diamond(\min_{\neg \mathbf{b}} \vee \min_{\neg \mathbf{b}}^{\leftarrow})$.

For formula (8) observe that, by conditions (i), (iii), and (iv) in Definition 6.4, every successor of r that satisfies $\min_{\neg \mathbf{b}}$ or $\min_{\mathbf{b}}^{\leftarrow}$ does not satisfy \mathbf{b} . Analogously, every successor of r that satisfies $\min_{\neg \mathbf{b}}^{\leftarrow}$ or $\min_{\mathbf{b}}$ satisfies \mathbf{b} .

For formula (9), let t and \bar{t} be arbitrary successors of r and \bar{r} , respectively. There are $k, i \in [0, \text{Tower}(\ell, n) - 1]$ such that the subtrees \mathfrak{T} and $\bar{\mathfrak{T}}$ rooted in t and \bar{t} , respectively are some $\Upsilon_{\ell-1,n}(k)$ and $\bar{\Upsilon}_{\ell-1,n}(i)$. Now, assume that $\mathfrak{T} \times \bar{\mathfrak{T}} \models \text{succ}_{\ell-1,n}$. By Point (d) of Lemma 6.5 we have $i = k + 1$. We need to show that $\mathfrak{T} \times \bar{\mathfrak{T}} \models \bigwedge_{x \in \{\mathbf{b}, \neg \mathbf{b}\}} ((\min_x \vee \min_x^{\leftarrow}) \leftrightarrow \min_x^{\leftarrow})$. We only show it for $x = \neg \mathbf{b}$, because the case $x = \mathbf{b}$ can be proved analogously.

If $j = \text{Tower}(\ell + 1, n) - 1$, then by condition (v) of Definition 6.4, every successor of r (respectively, \bar{r}) is labelled with $\min_{\neg b}^{\leftarrow}$ (resp., $\overline{\min_{\neg b}^{\leftarrow}}$). Hence, $\mathfrak{T} \times \bar{\mathfrak{T}} \models (\overline{\min_{\neg b}} \vee \overline{\min_{\neg b}^{\leftarrow}}) \leftrightarrow \min_{\neg b}^{\leftarrow}$ holds.

Now, assume that $j < \text{Tower}(\ell + 1, n) - 1$. Put $m = \text{Tower}(\ell, n) - 1$ and let t_0, \dots, t_m be the successors of r such that the subtree \mathfrak{T}_k rooted in t_k is some $\Upsilon_{\ell-1,n}(k)$ for all $k \in [0, m]$. Analogously, define $\bar{t}_0, \dots, \bar{t}_m$ to be the successors of \bar{r} such that the subtree $\bar{\mathfrak{T}}_k$ rooted in \bar{t}_k is some $\bar{\Upsilon}_{\ell-1,n}(k)$ for all $k \in [0, m]$. We have that $t = t_i$ and $\bar{t} = \bar{t}_{i+1}$. By conditions (i), (iii), and (v) of Definition 6.4, there is some $m_0 \in [0, m]$ such that

- $\mathfrak{T}_{m_0} \models \min_{\neg b}$ and $\bar{\mathfrak{T}}_{m_0} \models \overline{\min_{\neg b}}$,
- $\mathfrak{T}_k \models \min_{\neg b}^{\leftarrow}$ and $\bar{\mathfrak{T}}_k \models \overline{\min_{\neg b}^{\leftarrow}}$ for all $k < m_0$, and
- $\mathfrak{T}_k \models \neg \min_{\neg b} \wedge \neg \min_{\neg b}^{\leftarrow}$ and $\bar{\mathfrak{T}}_k \models \neg \overline{\min_{\neg b}} \wedge \neg \overline{\min_{\neg b}^{\leftarrow}}$ for all $k > m_0$.

Now, it is easy to verify that $\mathfrak{T} \times \bar{\mathfrak{T}} = \mathfrak{T}_i \times \bar{\mathfrak{T}}_{i+1} \models (\overline{\min_{\neg b}} \vee \overline{\min_{\neg b}^{\leftarrow}}) \leftrightarrow \min_{\neg b}^{\leftarrow}$.

“Only-if”: We prove also the “only-if” direction by induction on ℓ . For $\ell = 0$ assume $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{0,n}$. Thus, both s and \bar{s} do not have any successors and due to point (a) of Lemma 6.5 there exists some $j \in [0, \text{Tower}(1, n) - 1]$ such that (\mathfrak{S}, s) is bisimilar to some $\Upsilon_{0,n}(j)$ and $(\bar{\mathfrak{S}}, \bar{s})$ is bisimilar to some $\bar{\Upsilon}_{0,n}(j)$.

For the induction step, let us assume $\ell \geq 1$ and $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \varphi_{\ell,n}$ and put $m = \text{Tower}(\ell, n) - 1$.

Claim 1. For each successor t of s we have that (\mathfrak{S}, t) is bisimilar to some $\Upsilon_{\ell-1,n}(i)$ for some $i \in [0, m]$ and for each $i \in [0, m]$ there is a successor s_i of s such that (\mathfrak{S}, s_i) is bisimilar to some $\Upsilon_{\ell-1,n}(i)$. The analogous property holds for $\bar{\mathfrak{S}}$.

Proof of Claim 1. Let t be an arbitrary successor of s . By formula (2), there is a successor \bar{t} of t such that $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle t, \bar{t} \rangle) \models \varphi_{\ell-1,n}$. By induction hypothesis, there is a $i \in [0, m]$ such that (\mathfrak{S}, t) is bisimilar to some $\Upsilon_{\ell-1,n}(i)$ (and $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to some $\bar{\Upsilon}_{\ell-1,n}(i)$). Analogous reasoning for formula (3) yields that for every successor \bar{t} of \bar{s} there is some $i \in [0, m]$ such that $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to some $\bar{\Upsilon}_{\ell-1,n}(i)$.

Moreover, formulas (3), (4), and (5) imply that there are successors s_0, \dots, s_m of s and $\bar{s}_0, \dots, \bar{s}_m$ of \bar{s} such that (\mathfrak{S}, s_i) is bisimilar to some $\Upsilon_{\ell-1,n}(i)$ and $(\bar{\mathfrak{S}}, \bar{s}_i)$ is bisimilar to some $\bar{\Upsilon}_{\ell-1,n}(i)$: By formula (4) and point (b) of Lemma 6.5, there are such s_0, \bar{s}_0 . By formula (5) and points (c) and (d) of Lemma 6.5, there is such an \bar{s}_1 . By formula (3) (and reasoning as above), there is such an s_1 . Inductively repeating the argument yields the claimed s_0, \dots, s_m and $\bar{s}_0, \dots, \bar{s}_m$. This proves Claim 1.

Claim 2. If t, t' are successors of s that are bisimilar to $\Upsilon_{\ell-1,n}(i, V)$ and $\Upsilon_{\ell-1,n}(i, V')$ for some $i \in [0, m]$, then $(\mathfrak{S}, t) \sim (\mathfrak{S}, t')$. The analogous property holds for $(\bar{\mathfrak{S}})$.

Proof of Claim 2. It suffices to show that (\mathfrak{S}, t) and (\mathfrak{S}, t') satisfy the same propositions from \mathbf{P}_n . By applying formula (6) twice, we have for each $p \in \mathbf{P}_n$ that $(\mathfrak{S}, t) \models p$ iff $(\bar{\mathfrak{S}}, \bar{s}_i) \models \bar{p}$ iff $(\mathfrak{S}, t') \models p$. This proves Claim 2 as it shows $V = V'$.

By Claims 1 and 2, it is well-defined to set

$$I^+ = \{i \in [0, m] \mid (\mathfrak{S}, s_i) \models \mathbf{b}\} \quad \text{and} \quad j = \sum_{i \in I^+} 2^i.$$

Let moreover $V \subseteq P_n$ be the set of all propositions such that $(\mathfrak{S}, s) \models p$, and define \bar{V} analogously. By formula (1), V does not contain propositional variables \mathbf{b}_i for $i \in [0, n-1]$. Thus, we can finally define $(\mathfrak{T}, t) = \Upsilon_{\ell, n}(j, V)$ and $(\bar{\mathfrak{T}}, \bar{t}) = \Upsilon_{\ell, n}(j, \bar{V})$.

Claim 3. $(\mathfrak{T}, t) \sim (\mathfrak{S}, s)$ and $(\bar{\mathfrak{T}}, \bar{t}) \sim (\mathfrak{S}, \bar{s})$.

Proof of Claim 3. We only prove $(\mathfrak{T}, t) \sim (\mathfrak{S}, s)$ since the other case can be proved analogously. For each $i \in [0, m]$ define t_i to be the unique successor of t such that (\mathfrak{T}, t_i) is of the form $\Upsilon_{\ell-1, n}(i)$ and define $(\bar{\mathfrak{T}}, \bar{t}_i)$ analogously.

Note first that (\mathfrak{S}, s) and (\mathfrak{T}, t) satisfy the same atomic propositions (namely those from V) by definition of \mathfrak{T} . Thus, it remains to show the ‘back-and-forth’ condition of bisimulation, that is, for every successor s' of s we find a successor t' of t such that $(\mathfrak{S}, s') \sim (\mathfrak{T}, t')$, and vice versa. By Claims 1 and 2, it suffices to show for each $i \in [0, m]$ that $(\mathfrak{S}, s_i) \sim (\mathfrak{T}, t_i)$. As both (\mathfrak{S}, s_i) and (\mathfrak{T}, t_i) are bisimilar to some $\Upsilon_{\ell-1, n}(i)$, it remains to show $(\mathfrak{S}, s_i) \models p$ if and only if $(\mathfrak{T}, t_i) \models p$ for all $p \in P_n$.

By definition of I^+ , j , and since $(\mathfrak{T}, t) = \Upsilon_{\ell, n}(j, V)$, we surely have $(\mathfrak{S}, s_i) \models \mathbf{b}$ if and only if $(\mathfrak{T}, t_i) \models \mathbf{b}$. It is also not hard to verify that they agree on the propositions $\{\mathbf{b}_0, \dots, \mathbf{b}_{n-1}\}$: If $\ell > 1$, then both $(\mathfrak{S}, s_i) \not\models \mathbf{b}_k$ and $(\mathfrak{T}, t_i) \not\models \mathbf{b}_k$ for any $k \in [0, n-1]$; if $\ell = 1$, then (\mathfrak{S}, s_i) and (\mathfrak{T}, t_i) are $(0, n)$ -trees and i uniquely determines the interpretation of these propositions by Definition 6.4.

It remains to consider the propositional variables from $\{\min_{\neg \mathbf{b}}, \min_{\neg \mathbf{b}}^{\leftarrow}, \min_{\mathbf{b}}, \min_{\mathbf{b}}^{\leftarrow}\}$. We concentrate on the propositions $\min_{\neg \mathbf{b}}$ and $\min_{\neg \mathbf{b}}^{\leftarrow}$ since for the others it can be proved analogously. First, note that we have the following:

- (a) For each $i \in [0, m]$ we have that (\mathfrak{S}, s_i) does not satisfy both $\min_{\neg \mathbf{b}}$ and $\min_{\neg \mathbf{b}}^{\leftarrow}$ since otherwise this would imply $(\mathfrak{S}, s_i) \models \mathbf{b} \wedge \neg \mathbf{b}$ by formula (8), a contradiction.
- (b) If there exists some $i_0 \in [0, m]$ such that $(\mathfrak{S}, s_{i_0}) \models \min_{\neg \mathbf{b}} \vee \min_{\neg \mathbf{b}}^{\leftarrow}$, then $(\mathfrak{S}, s_i) \models \min_{\neg \mathbf{b}}^{\leftarrow}$ for all $i \in [0, i_0 - 1]$: By formula (6), we have $(\bar{\mathfrak{S}}, \bar{s}_{i_0}) \models \overline{\min_{\neg \mathbf{b}} \vee \min_{\neg \mathbf{b}}^{\leftarrow}}$. By formula (9), we obtain $(\mathfrak{S}, s_{i_0-1}) \models \min_{\neg \mathbf{b}}^{\leftarrow}$. This argument can be continued inductively.

As $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle)$ satisfies formula (7) one of the following cases has to appear:

Case 1. There exists some $i_0 \in [0, m]$ such that $(\mathfrak{S}, s_{i_0}) \models \min_{\neg \mathbf{b}}$. We observe:

- (c) $(\mathfrak{S}, s_{i_0}) \models \neg \mathbf{b}$ by formula (8);
- (d) $(\mathfrak{S}, s_i) \models \min_{\neg \mathbf{b}}^{\leftarrow}$ for each $i \in [0, i_0 - 1]$ by point (b) above and hence $(\mathfrak{S}, s_i) \models \mathbf{b}$ by formula (8) for each $i \in [0, i_0 - 1]$;

- (e) $(\mathfrak{S}, s_i) \not\models (\min_{\neg b}^{\leftarrow} \vee \min_{\neg b})$ for each $i \in [i_0 + 1, m]$. Assume the contrary, namely $(\mathfrak{S}, s_i) \models (\min_{\neg b}^{\leftarrow} \vee \min_{\neg b})$ for some $i \in [i_0 + 1, m]$. By point (b) above, we get in particular $(\mathfrak{S}, s_{i_0}) \models \min_{\neg b}^{\leftarrow}$, which is in contradiction to point (a).

By comparing the above points (a) to (e) with conditions (i)-(v) of Definition 6.4, one sees that (\mathfrak{S}, s_i) and (\mathfrak{T}, t_i) satisfy the same propositions from $\{\min_{\neg b}, \min_{\neg b}^{\leftarrow}\}$ for each $i \in [0, m]$.

Case 2. There is no $i \in [0, m]$ such that $(\mathfrak{S}, s_i) \models \min_{\neg b}$. We observe:

- (c') There exists some $i_0 \in [0, m]$ such that $(\mathfrak{S}, s_{i_0}) \models \min_{\neg b}^{\leftarrow}$ by formula (7) and hence $(\mathfrak{S}, s_{i_0}) \models b$ by formula (8);
- (d') $(\mathfrak{S}, s_i) \models \min_{\neg b}^{\leftarrow}$ for each $i \in [0, i_0 - 1]$ by point (b) above and hence $(\mathfrak{S}, s_i) \models b$ by formula (8) for each $i \in [0, i_0 - 1]$;
- (e') $(\mathfrak{S}, s_i) \models \min_{\neg b}^{\leftarrow}$ and thus $(\mathfrak{S}, s_i) \models b$ by formula (8) for each $i \in [i_0 + 1, m]$. This is proved by repeatedly applying the following three steps:
 - (i) By formula (9), $(\mathfrak{S}, s_{i_0}) \models \min_{\neg b}^{\leftarrow}$ implies $(\overline{\mathfrak{S}}, \bar{s}_{i_0+1}) \models \overline{\min_{\neg b}} \vee \overline{\min_{\neg b}^{\leftarrow}}$.
 - (ii) Assume $(\overline{\mathfrak{S}}, \bar{s}_{i_0+1}) \models \overline{\min_{\neg b}}$. By formula (6), we have $(\mathfrak{S}, s_{i_0+1}) \models \min_{\neg b}$, contradicting the assumption for Case 2. Thus, $(\overline{\mathfrak{S}}, \bar{s}_{i_0+1}) \models \overline{\min_{\neg b}^{\leftarrow}}$.
 - (iii) By formula (6), we have $(\mathfrak{S}, s_{i_0+1}) \models \min_{\neg b}^{\leftarrow}$.

By comparing the above points (a), (b), and (c') to (e') with conditions (i)-(v) of Definition 6.4 one sees that \mathfrak{S}_i and \mathfrak{T}_i satisfy the same propositions from $\{\min_{\neg b}, \min_{\neg b}^{\leftarrow}\}$ for each $i \in [0, m]$. This concludes the proof Claim 3 and thus of Theorem 6.7. \blacksquare

6.2.3 ℓ -NEXPTIME-hardness for each $\ell \geq 1$

We are now ready to proceed to the main result of Section 6.2. By making use of the formulas $\varphi_{\ell, n}$, we can encode big numbers. In the proof of the following proposition we use these numbers to encode (the coordinates of) big tiling problems. Recall that ℓ -NEXPTIME = $\bigcup_{k \geq 0} \text{NTIME}(\text{Tower}(\ell, n^k))$ for $\ell \geq 0$.

Theorem 6.8. *The following holds:*

- For each $\ell \geq 1$, \mathbf{K}_{id}^2 -SAT restricted to formulas of switching depth ℓ is ℓ -NEXPTIME-hard under polynomial time many-one reductions.
- In particular, \mathbf{K}_{id}^2 -SAT is nonelementary.

For the proof of Theorem 6.8, we need to introduce tilings and tiling problems. A *tiling system* is a tuple $S = (\Theta, \mathbb{H}, \mathbb{V})$, where Θ is a finite set of *tile types*, $\mathbb{H} \subseteq \Theta \times \Theta$ is the *horizontal matching relation*, and $\mathbb{V} \subseteq \Theta \times \Theta$ is the *vertical matching relation*. A mapping $\tau : [0, k - 1] \times [0, k - 1] \rightarrow \Theta$ for $k \geq 0$ is a *k-solution* for S if for all $(x, y) \in [0, k - 1] \times [0, k - 1]$ the following holds:

- if $x < k - 1$, $\tau(x, y) = \theta$, and $\tau(x + 1, y) = \theta'$, then $(\theta, \theta') \in \mathbb{H}$, and
- if $y < k - 1$, $\tau(x, y) = \theta$, and $\tau(x, y + 1) = \theta'$, then $(\theta, \theta') \in \mathbb{V}$.

Let $w = \theta_0 \cdots \theta_{n-1} \in \Theta^n$ be a word and let $k \geq n$. With $\text{Sol}_k(S, w)$ we denote the set of all k -solutions τ for S such that $\tau(x, 0) = \theta_x$ for all $x \in [0, n - 1]$. For a tiling system $S = (\Theta, \mathbb{H}, \mathbb{V})$ we define its ℓ -EXP-tiling problem as follows:

ℓ -EXP-TILING PROBLEM FOR $S = (\Theta, \mathbb{H}, \mathbb{V})$

INPUT: A word $w \in \Theta^n$.

OUTPUT: Does $\text{Sol}_{\text{Tower}(\ell, n)}(S, w) \neq \emptyset$ hold?

The following result is well-known, see for instance [22, 30].

Theorem 6.9. *For each $\ell \geq 1$, there exists a fixed tiling system S_ℓ such that the ℓ -EXP-TILING PROBLEM FOR S_ℓ is hard for ℓ -NEXPTIME under polynomial time many-one reductions.*

The proof of this theorem is based on the observation that from a nondeterministic $t(n)$ -time bounded Turing machine M one can construct a tiling system S_M which simulates M in the following sense: from an input w of length n for M one can construct a word x_w of length n over the tile types of S_M such that M accepts w if and only if $\text{Sol}_{t(n)}(S_M, x_w) \neq \emptyset$. Intuitively, in a $t(n)$ -solution τ for S_M each row, i.e., the sequence $\tau(0, i), \dots, \tau(t(n) - 1, i)$ for some i , encodes a configuration. The horizontal matching relation of S_M ensures that each row is a valid configuration, while the vertical matching relation guarantees valid transitions between configurations.

We can finally prove Theorem 6.8. We give the detailed proof only for $\ell \geq 2$ and just remark that for $\ell = 1$ it is a straightforward adaption of Definition 6.6 and Theorem 6.7. The structure of the proof (for $\ell \geq 2$) is as follows:

- In the first step, we define structures, called *grid element trees* representing one particular cell, i.e., a triple (X, Y, θ) of two coordinates X, Y , and a tile type θ , of the $\text{Tower}(\ell, n)$ -solution. Moreover, we provide a formula **gridel** enforcing (again: up to bisimulation equivalence) such grid element trees.
- In the second step, we define structures, called *tiling trees* representing a complete $\text{Tower}(\ell, n)$ -solution for some fixed tiling system S .
- Finally, we give a formula **tiling** that enforces tiling trees (up to bisimulation) and a formula φ_w that deals with the input w of length n . In particular, **tiling** \wedge φ_w will be satisfiable if and only if $\text{Sol}_{\text{Tower}(\ell, n)}(S, w)$ is not empty.

The proof reuses ideas of the proofs of Lemma 6.5 and Theorem 6.7, so we will not give all details. However, all missing details are straightforward variants of the mentioned proofs.

Proof of Theorem 6.8. Fix some $\ell \geq 2$ and let $S_\ell = (\Theta, \mathbb{H}, \mathbb{V})$ be some tiling system such that the ℓ -EXP-TILING PROBLEM FOR S_ℓ is hard for ℓ -NEXPTIME (exists by Theorem 6.9). We give a polynomial time many-one reduction from the ℓ -EXP-TILING PROBLEM FOR S_ℓ to \mathbf{K}_{id}^2 -SAT restricted to formulas of switching depth ℓ . Let $w = \theta_0 \cdots \theta_{n-1}$ be an input word for the ℓ -EXP-tiling problem for S_ℓ and let $m = \text{Tower}(\ell - 1, n) - 1$.

We add to the set of propositions \mathbf{P}_n from the previous section all tile types from $\theta \in \Theta$ and two additional propositions x and y , and analogously, we add propositions $\bar{\theta}$ for $\theta \in \Theta$, \bar{x} , and \bar{y} to $\bar{\mathbf{P}}_n$. The sets $\mathbf{A} = \{a\}, \bar{\mathbf{A}} = \{\bar{a}\}$ remain unchanged. For $z \in \{x, y\}$ we define $\Upsilon_{\ell-1,n}^{(1,z)}(j)$ as the tree $\Upsilon_{\ell-1,n}(j, \{\mathbf{b}, z\})$ and $\Upsilon_{\ell-1,n}^{(0,z)}(j)$ as $\Upsilon_{\ell-1,n}(j, \{z\})$. In the same way define $\bar{\Upsilon}_{\ell-1,n}^{(\beta,z)}(j)$.

Now, assume $X, Y \in [0, \text{Tower}(\ell, n) - 1]$ and $\theta \in \Theta$. Note that X, Y uniquely determine sets $I_x, I_y \subseteq [0, m]$ by

$$X = \sum_{i \in I_x} 2^i \quad \text{and} \quad Y = \sum_{i \in I_y} 2^i.$$

Then the *grid element tree* $\mathfrak{G}(X, Y, \theta)$ is obtained as follows:

- Take the disjoint union of a root node r and all trees from the set

$$\begin{aligned} U = & \{\Upsilon_{\ell-2,n}^{(1,x)}(i) \mid i \in I_x\} \cup \{\Upsilon_{\ell-2,n}^{(0,x)}(i) \mid i \notin I_x\} \cup \\ & \{\Upsilon_{\ell-2,n}^{(1,y)}(i) \mid i \in I_y\} \cup \{\Upsilon_{\ell-2,n}^{(0,y)}(i) \mid i \notin I_y\}. \end{aligned}$$

- Add an a -transition from the root r to the root of each tree from U .
- Label the root r with θ .

For an example grid element tree, see Figure 6.4. The tree $\bar{\mathfrak{G}}(X, Y, \theta)$ is obtained from $\mathfrak{G}(X, Y, \theta)$ by replacing every accessibility relation \xrightarrow{a} by $\xrightarrow{\bar{a}}$ and every proposition p by \bar{p} .

In order to enforce grid element trees, we need to slightly modify the formulas used in the proof of Theorem 6.7. For this purpose, it is useful to have for $z \in \{x, y\}$ the abbreviations $\Diamond_z \psi = \Diamond(z \wedge \psi)$, $\Box_z \psi = \Box(z \rightarrow \psi)$, $\bar{\Diamond}_z \psi = \bar{\Diamond}(\bar{z} \wedge \psi)$, and $\bar{\Box}_z \psi = \bar{\Box}(\bar{z} \rightarrow \psi)$. Then, for $z \in \{x, y\}$ we can define relativized formulas $\varphi_{\ell-1,n}^z$, $\text{eq}_{\ell-1,n}^z$, $\text{first}_{\ell-1,n}^z$, $\text{last}_{\ell-1,n}^z$, and $\text{succ}_{\ell-1,n}^z$ by replacing in the definitions of the formulas $\varphi_{\ell-1,n}$, $\text{eq}_{\ell-1,n}$, $\text{first}_{\ell-1,n}$, $\text{last}_{\ell-1,n}$, and $\text{succ}_{\ell-1,n}$ every modality \Diamond (resp., \Box , $\bar{\Diamond}$, $\bar{\Box}$) by \Diamond_z (resp., \Box_z , $\bar{\Diamond}_z$, $\bar{\Box}_z$). All occurrences of $\varphi_{\ell-2,n}$, $\text{eq}_{\ell-2,n}$, $\text{first}_{\ell-2,n}$, $\text{last}_{\ell-2,n}$, and $\text{succ}_{\ell-2,n}$ are not changed, i.e., we do not replace modalities within these formulas. The following Claim can be verified along the lines of the proof of Lemma 6.5.

Claim 1. Let $N_x, N_y, \bar{N}_x, \bar{N}_y \in [0, \text{Tower}(\ell, n) - 1]$ and $\theta, \theta' \in \Theta$ and let $\mathfrak{T} = \mathfrak{G}(N_x, N_y, \theta)$ and $\bar{\mathfrak{T}} = \bar{\mathfrak{G}}(\bar{N}_x, \bar{N}_y, \theta')$ be grid element trees. Then the following holds for all $z \in \{x, y\}$:

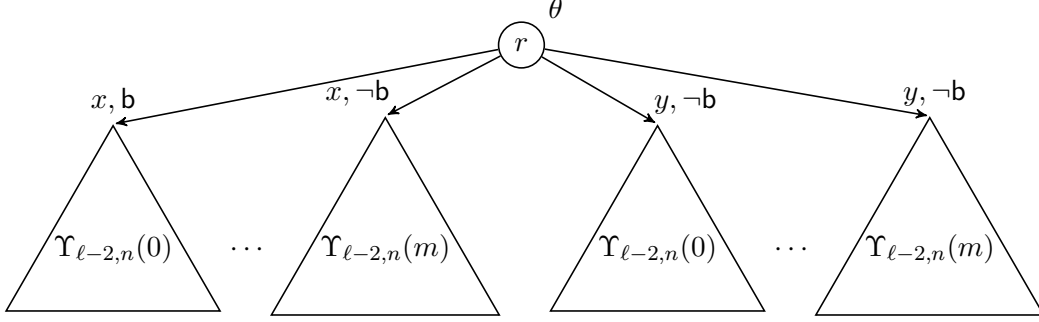


Figure 6.4: Example grid element tree. The root r has $m + 1$ successors labeled with x and $m + 1$ successors labeled with y ; hence, the grid element tree encodes *two* numbers X and Y .

- (a) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{eq}_{\ell-1,n}^z$ if and only if $N_z = \overline{N_z}$.
- (b) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{first}_{\ell-1,n}^z$ if and only if $N_z = \overline{N_z} = 0$.
- (c) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{last}_{\ell-1,n}^z$ if and only if $N_z = \overline{N_z} = \text{Tower}(\ell, n) - 1$.
- (d) $\mathfrak{T} \times \overline{\mathfrak{T}} \models \text{succ}_{\ell-1,n}^z$ if and only if $\overline{N_z} = N_z + 1$.

Using the relativized version of $\varphi_{\ell-1,n}$ we can enforce grid element trees. We define **gridel** as the conjunction of

$$\bigvee_{\theta \in \Theta} (\theta \wedge \overline{\theta} \wedge \bigwedge_{\kappa \in \Theta \setminus \{\theta\}} (\neg \kappa \wedge \neg \overline{\kappa})) \wedge \square \square ((x \oplus y) \wedge (\overline{x} \oplus \overline{y})) \wedge \varphi_{\ell-1,n}^x \wedge \varphi_{\ell-1,n}^y,$$

and

$$\bigwedge_{p \in P_n \setminus \Theta} (\neg p \wedge \neg \overline{p}) \wedge \bigwedge_{2 \leq i \leq \ell-1} \bigwedge_{p \in \Theta \cup \{x,y\}} \square^i \square^i (\neg p \wedge \neg \overline{p})$$

where \oplus denotes “exclusive or” and \square^i denotes the sequence of i boxes \square . Intuitively, the first formula expresses that (i) the root is labeled with precisely one symbol $\theta \in \Theta$ and (ii) we can associate precisely two values with the grid element structure: the value enforced by $\varphi_{\ell-1,n}^x$ (in analogy to Theorem 6.7) and the value enforced by $\varphi_{\ell-1,n}^y$. Additionally, all successor worlds are labeled with either x or y , so the formula $\varphi_{\ell-1,n}^x, \varphi_{\ell-1,n}^y$ really determine all successor worlds. The second formula is just an auxiliary formula restricting the newly introduced propositions appropriately, similar to formula (1) of Definition 6.6. The following claim makes this property of **gridel** explicit.

Claim 2. For all structures (\mathfrak{S}, s) and $(\overline{\mathfrak{S}}, \overline{s})$ we have that $(\mathfrak{S} \times \overline{\mathfrak{S}}, \langle s, \overline{s} \rangle) \models \text{gridel}$ if and only if there are $X, Y \in [0, \text{Tower}(\ell, n) - 1]$ and $\theta \in \Theta$ such that (\mathfrak{S}, s) and $(\overline{\mathfrak{S}}, \overline{s})$ are bisimilar to grid element structures $\mathfrak{G}(X, Y, \theta)$ and $\overline{\mathfrak{G}}(X, Y, \theta)$, respectively.

Next, let $\tau : [0, \text{Tower}(\ell, n) - 1]^2 \rightarrow \Theta$ be a mapping. We define the *tiling tree* $\mathfrak{T}(\tau)$ as follows:

- Take the disjoint union of a root node r and all grid element trees from the set

$$U = \{\mathfrak{G}(X, Y, \tau(X, Y)) \mid X, Y \in [0, \text{Tower}(\ell, n) - 1]\}.$$

- Add an edge from the root r to the root of each tree from U .

Intuitively, a tiling tree $\mathfrak{T}(\tau)$ represents the mapping τ as follows: for every X, Y in the domain of τ it has a successor that is a grid element tree encoding the triple $(X, Y, \tau(X, Y))$. The copy $\overline{\mathfrak{T}}(\tau)$ is defined as usual. The following claim states the existence of a formula that enforces tiling trees.

Claim 3. There is a formula **tiling** of switching depth ℓ such that for all pointed structures (\mathfrak{S}, s) and $(\overline{\mathfrak{S}}, \bar{s})$ we have $(\mathfrak{S} \times \overline{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \text{tiling}$ if and only if there is a $\text{Tower}(\ell, n)$ -solution τ of S_ℓ such that (\mathfrak{S}, s) is bisimilar to the tiling tree $\mathfrak{T}(\tau)$ and $(\overline{\mathfrak{S}}, \bar{s})$ is bisimilar to $\overline{\mathfrak{T}}(\tau)$.

Proof of Claim 3. We take for **tiling** the conjunction of the following formulas:

$$(1) \bigwedge_{p \in P_n} (\neg p \wedge \neg \bar{p})$$

$$(2) \Diamond \overline{\Diamond}(\text{gridel} \wedge \text{first}_{\ell-1, n}^x \wedge \text{first}_{\ell-1, n}^y)$$

$$(3) \Box \overline{\Diamond} \text{gridel}$$

$$(4) \overline{\Box} \Diamond \text{gridel}$$

$$(5) \overline{\Box} \Box ((\text{eq}_{\ell-1, n}^x \wedge \text{eq}_{\ell-1, n}^y) \rightarrow \bigwedge_{\theta \in \Theta} (\theta \leftrightarrow \bar{\theta}))$$

$$(6) \Box (\overline{\Box} \neg \text{last}_{\ell-1, n}^x \rightarrow \overline{\Diamond} (\text{succ}_{\ell-1, n}^x \wedge \text{eq}_{\ell-1, n}^y \wedge \bigvee_{(\theta_1, \theta_2) \in \mathbb{H}} (\theta_1 \wedge \bar{\theta}_2)))$$

$$(7) \Box (\overline{\Box} \neg \text{last}_{\ell-1, n}^y \rightarrow \overline{\Diamond} (\text{succ}_{\ell-1, n}^y \wedge \text{eq}_{\ell-1, n}^x \wedge \bigvee_{(\theta_1, \theta_2) \in \mathbb{V}} (\theta_1 \wedge \bar{\theta}_2)))$$

To show that the formula **tiling** satisfies the statements from the Claim we proceed similarly as in the proof of Theorem 6.7. Observe first that **tiling** has switching depth ℓ . For the “if”-direction of the statement assume a $\text{Tower}(\ell, n)$ -solution τ and structures (\mathfrak{S}, s) and $(\overline{\mathfrak{S}}, \bar{s})$ that are bisimilar to $\mathfrak{T}(\tau)$ and $\overline{\mathfrak{T}}(\tau)$, respectively. It is routine to verify that $(\mathfrak{S} \times \overline{\mathfrak{S}}, \langle s, \bar{s} \rangle)$ satisfies all the formulas (1)-(7) given above.

For the “only-if”-direction assume that $(\mathfrak{S} \times \overline{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \text{tiling}$. Formulas (3) and (4) enforce that for all successors t and \bar{t} of s and \bar{s} , respectively, we have that (\mathfrak{S}, t) is

bisimilar to some grid element tree $\mathfrak{G}(X, Y, \theta)$ and $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to some grid element tree $\bar{\mathfrak{G}}(X, Y, \theta)$. Formula (2) enforces the existence of successors t and \bar{t} such that (\mathfrak{S}, t) is bisimilar to the grid element tree $\mathfrak{G}(0, 0, \theta)$ and $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to the grid element tree $\bar{\mathfrak{G}}(0, 0, \theta')$ for some $\theta, \theta' \in \Theta$. Starting from this $\mathfrak{G}(0, 0, \theta)$, formulas (6), (7), and (4) inductively enforce the existence of grid element trees $\mathfrak{G}(i, j, \theta)$ for each $i, j \in [0, \text{Tower}(\ell, n) - 1]$ and some θ . Assume a successor t of s such that (\mathfrak{S}, t) is bisimilar to the grid element tree $\mathfrak{G}(i, j, \theta)$. If $i < \text{Tower}(\ell, n) - 1$, then formula (6) enforces the existence of a successor \bar{t} of \bar{s} such that $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to a grid element tree $\bar{\mathfrak{G}}(i + 1, j, \theta')$ with $(\theta, \theta') \in \mathbb{H}$. Likewise, if $j < \text{Tower}(\ell, n) - 1$, then formula (7) enforces the existence of a successor \bar{t} of \bar{s} such that $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to a grid element tree $\bar{\mathfrak{G}}(i, j + 1, \theta')$ with $(\theta, \theta') \in \mathbb{V}$. In both cases, by formula (4), there is some successor s' of s such that (\mathfrak{S}, s') is bisimilar to $\mathfrak{G}(i + 1, j, \theta')$ or $\mathfrak{G}(i, j + 1, \theta')$, respectively.

Thus, for each $i, j \in [0, \text{Tower}(\ell, n) - 1]$ there are worlds t, \bar{t} , and a tile type $\theta \in \Theta$ such that

- t is a successor of s and \bar{t} is a successor of \bar{s} ;
- (\mathfrak{S}, t) is bisimilar to the grid element tree $\mathfrak{G}(i, j, \theta)$;
- $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to the grid element tree $\bar{\mathfrak{G}}(i, j, \theta)$.

Assume now that there are successors t, t' of s such that (\mathfrak{S}, t) is bisimilar to $\mathfrak{G}(i, j, \theta_1)$ and (\mathfrak{S}, t') is bisimilar to $\mathfrak{G}(i, j, \theta_2)$. By the above, there is a successor \bar{t} of \bar{s} such that $(\bar{\mathfrak{S}}, \bar{t})$ is bisimilar to $\bar{\mathfrak{G}}(i, j, \theta')$ for some θ' . By Claim 1, we have $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle t, \bar{t} \rangle) \models \text{eq}_{\ell-1, n}^x \wedge \text{eq}_{\ell-1, n}^y$. By formula (5), we obtain $\theta' = \theta_1$. Analogously, we get $\theta' = \theta_2$ and thus $\theta_1 = \theta_2$. Hence, for every $i, j \in [0, \text{Tower}(\ell, n) - 1]$ there is a *unique* θ_{ij} such that for all successors t of s with (\mathfrak{S}, t) bisimilar to $\mathfrak{G}(i, j, \theta')$ we have $\theta' = \theta_{ij}$ (and analogously, for all successors \bar{t} of \bar{s}). Thus, the mapping τ defined by $\tau(i, j) = \theta_{ij}$ is well-defined. Moreover, by construction it is a $\text{Tower}(\ell, n)$ -solution for S_ℓ .

We claim that (\mathfrak{S}, s) and $(\bar{\mathfrak{S}}, \bar{s})$ are bisimilar to the tiling trees $\mathfrak{T}(\tau)$ and $\bar{\mathfrak{T}}(\tau)$, respectively. By the properties observed above, it suffices to note that the (\mathfrak{S}, s) and $(\bar{\mathfrak{S}}, \bar{s})$ do not satisfy any propositions from P_n by formula (1) as required by the definition of a tiling tree. This finishes the proof of Claim 3.

Finally, it is straightforward to write down a formula φ_w expressing that $\tau(i, 0) = \theta_i$ for all $i \in [0, n - 1]$. We can for each $i \in [0, n - 1]$ give a formula val_i such that for all structures (\mathfrak{S}, s) , $(\bar{\mathfrak{S}}, \bar{s})$ that are bisimilar to grid element structures $\mathfrak{G}(X, Y, \theta)$ and $\bar{\mathfrak{G}}(X', Y', \theta')$, respectively, we have that $(\mathfrak{S} \times \bar{\mathfrak{S}}, \langle s, \bar{s} \rangle) \models \text{val}_i$ if and only if $X = X' = i$. Hence, we can define

$$\varphi_w = \bigwedge_{i=0}^{n-1} \Box \Box (\text{val}_i \wedge \text{first}_{\ell-1, n}^y \rightarrow \theta_i).$$

Hence, $\text{Sol}_{\text{Tower}(\ell, n)}(S_\ell, w) \neq \emptyset$ if and only if $\text{tiling} \wedge \varphi_w$ is interpreted satisfiable. It remains to note that the size of the formula φ_w is polynomial in the size of w . This concludes the proof. \blacksquare

The following corollary is an immediate consequence of Proposition 6.2 and Theorem 6.8.

Corollary 6.10. *The following holds:*

- For each $\ell \geq 1$, \mathbf{K}^2 -SAT restricted to formulas of switching depth ℓ is ℓ -NEXPTIME-hard under polynomial time many-one reductions.
- In particular, \mathbf{K}^2 -SAT is nonelementary.

As a final remark let us mention that there is a close connection of our lower bound technique and the decision procedure devised in [105]. In a nutshell, it is a filtration algorithm that factors through the set $D_m(\mathbf{P})$ of (\mathbf{A}, \mathbf{P}) -formulas of modal depth at most m . Clearly, there are infinitely such formulas but it is well-known that (i) there are finitely (more precisely: $\text{Tower}(m, |\mathbf{P}|)$) many *up to equivalence* and (ii) each equivalence class can be described by a tree of depth at most m and having an outdegree of $\text{Tower}(m, |\mathbf{P}|)$ in the worst case. In fact, they claim that their decision procedure can actually be strengthened to filter only through $D_\ell(\mathbf{P})$ where ℓ is the *switching depth* of some formula φ [105] and our result shows that there is no better way than that.

6.3 Hardness results for $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$

In this section, we prove further nonelementary lower bound results for the satisfiability problem of two-dimensional modal logics on restricted classes of frames. We hereby close nonelementary complexity gaps that were stated as open problems in [55]. Again, we exploit Proposition 6.2 and focus on satisfiability in *interpreted* product structures. Note that this is possible since Proposition 6.2 does not make any assumption on the underlying frames and thus holds also for restricted frame classes. Let us define the following logics:

- **$\mathbf{K4} \times \mathbf{K}$** : Two-dimensional modal logic restricted to products based on frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ where \mathfrak{F}_1 is a frame (W, \rightarrow_a) such that \rightarrow_a is *transitive*.
- **$\mathbf{S4} \times \mathbf{K}$** : Two-dimensional modal logic restricted to products based on frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ where \mathfrak{F}_1 is a frame (W, \xrightarrow{a}) such that \rightarrow_a is *transitive* and *reflexive*.
- **$\mathbf{S5}_2 \times \mathbf{K}$** : Two-dimensional modal logic restricted to products based on frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ where \mathfrak{F}_1 is a frame (W, \equiv, \approx) with equivalence relations \equiv and \approx .

We lift the terminology regarding satisfiability from the previous previous section. For instance, we say that an (\mathbf{A}, \mathbf{P}) -formula φ is uninterpreted (interpreted) satisfiable in, say, $\mathbf{K4} \times \mathbf{K}$ if there is an uninterpreted (interpreted) product model of φ with underlying frame $\mathfrak{F}_1 \times \mathfrak{F}_2$ and \mathfrak{F}_1 transtive.

Note that the lower bounds from the last section already hold for formulas having only accessibility relations $\xrightarrow{a}, \xrightarrow{\bar{a}}$, i.e., one accessibility relation for each component. Hence, throughout this section we fix $\mathbf{A} = \{a, \bar{a}\}$ and some countable set $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2$ of propositions. As in the previous section we will abbreviate \Diamond_a with \Diamond and $\Diamond_{\bar{a}}$ with $\bar{\Diamond}$. The idea for all three logics is to *lift* existing reductions from satisfiability in (one-dimensional) \mathbf{K} to satisfiability in (one-dimensional) $\mathbf{K4}, \mathbf{S4}, \mathbf{S5}_2$, see [96, 69, 52].

Let us start with $\mathbf{K4} \times \mathbf{K}$; the case $\mathbf{S4} \times \mathbf{K}$ works analogously. The idea for the reduction is to introduce additional propositions h_0, \dots, h_r to simulate *levels* in the model. Intuitively, h_i will be true in some world s' precisely when s' is i transitions away from s , the world where the input formula is witnessed. More formally, let φ be an (\mathbf{A}, \mathbf{P}) -formula with $\text{depth}_1(\varphi) = r$ and let h_0, \dots, h_r be fresh propositions. For every $0 \leq k \leq r$, we specify by structural induction a translation function t_k such that t_k is defined for an input formula ψ whenever $\text{depth}_1(\psi) + k \leq r$. More precisely, we set

$$\begin{aligned} t_k(p) &= H_k \wedge p && \text{for all } p \in \mathbf{P} \\ t_k(\neg\psi) &= H_k \wedge \neg t_k(\psi) \\ t_k(\psi_1 \wedge \psi_2) &= t_k(\psi_1) \wedge t_k(\psi_2) \\ t_k(\bar{\Diamond}\psi) &= \bar{\Diamond} t_k(\psi) \\ t_k(\Diamond\psi) &= H_k \wedge \Diamond(H_{k+1} \wedge t_{k+1}(\psi)) \end{aligned}$$

where $H_k = h_k \wedge \bigwedge_{i \neq k} \neg h_i$ and $k < r$ in the definition of $t_k(\Diamond\psi)$. We show that the translation is satisfiability preserving. More precisely, we prove the following lemma.

Lemma 6.11. *For every (\mathbf{A}, \mathbf{P}) -formula φ we have: φ is interpreted satisfiable in \mathbf{K}^2 iff $t_0(\varphi)$ is interpreted satisfiable in $\mathbf{K4} \times \mathbf{K}$.*

Proof. We assume that φ is defined over $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2$ for disjoint \mathbf{P}_1 and \mathbf{P}_2 . Moreover set $r = \text{depth}_1(\varphi)$. As in Section 6.2 we will write $\mathfrak{S}_1 \times \mathfrak{S}_2$ for $\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2$.

Assume first that φ is interpreted satisfiable in $\mathbf{K} \times \mathbf{K}$. Thus, there are structures

$$\mathfrak{S}_i = (W_i, \longrightarrow_i, \{W_{i,p} \mid p \in \mathbf{P}_i\})$$

($i \in \{1, 2\}$) and $\bar{s} = \langle s_1, s_2 \rangle \in W_1 \times W_2$ such that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi$. Without loss of generality we assume that \mathfrak{S}_1 is a tree with root s_1 . Define

$$\mathfrak{S}'_1 = (W_1, \longrightarrow_1^+, \{W'_{1,p} \mid p \in \mathbf{P}_1 \cup \{h_0, \dots, h_r\}\}),$$

where

- \longrightarrow_1^+ is the transitive closure of \longrightarrow_1 ,

- $W'_{1,p} = W_{1,p}$ for all $p \in \mathbf{P}_1$, and
- $W'_{1,h_i} = V_i$, where V_i is defined to be the set of worlds s' such that the (unique) path in \mathfrak{S}_1 from s_1 to s' has length i , i.e., consists of i transitions.

Note that by construction of \mathfrak{S}'_1 , we have for all $\bar{x} \in W_1 \times W_2$, $k \in [0, r]$:

$$(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models H_k \quad \Leftrightarrow \quad \bar{x} \in V_k \times W_2. \quad (6.2)$$

We prove by induction on the structure of φ that for each subformula ψ it holds: for all $i \in [0, r]$ and all $\bar{x} \in V_{r-i} \times W_2$. we have

$$\text{depth}_1(\psi) \leq i \quad \Rightarrow \quad ((\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models \psi \quad \Leftrightarrow \quad (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models t_{r-i}(\psi)).$$

For the induction base, assume $\psi = p$ for some atomic proposition $p \in \mathbf{P}_1 \cup \mathbf{P}_2$, i arbitrary in $[0, r]$, and fix an arbitrary $\bar{x} = \langle x_1, x_2 \rangle \in V_{r-i} \times W_2$. The statement is then a direct consequence of (6.2).

For the induction step, assume ψ is not atomic and $i \in [0, r]$ such that $i \geq \text{depth}_1(\psi)$, and let us fix some $\bar{x} \in V_{r-i} \times W_2$. We make a case distinction on the structure of ψ . For the cases $\neg\chi$, $\chi_1 \wedge \chi_2$, and $\Diamond\chi$ the equivalence follows straightforwardly from (6.2) and the induction hypothesis since $\text{depth}_1(\psi) = \text{depth}_1(\chi)$ and $\text{depth}_1(\chi_i) \leq \text{depth}_1(\psi)$ for $i \in \{1, 2\}$.

It remains to consider the case $\psi = \Diamond\chi$. Then $\text{depth}_1(\chi) = \text{depth}_1(\psi) - 1 \leq i - 1$ and we have that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models \psi$ is equivalent to the following:

$$\begin{aligned} & \exists \bar{y} \in V_{r-(i-1)} \times W_2 : \bar{x} \longrightarrow_1 \bar{y} \text{ and } (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{y}) \models \chi \\ \stackrel{\text{IH}}{\Leftrightarrow} & \exists \bar{y} \in V_{r-i+1} \times W_2 : \bar{x} \longrightarrow_1 \bar{y} \text{ and } (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{y}) \models t_{r-i+1}(\chi) \\ \stackrel{(6.2)}{\Leftrightarrow} & \exists \bar{y} \in V_{r-i+1} \times W_2 : \bar{x} \longrightarrow_1^+ \bar{y} \text{ and } (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{y}) \models H_{r-i+1} \wedge t_{r-i+1}(\chi) \\ \Leftrightarrow & (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models \Diamond(H_{r-i+1} \wedge t_{r-i+1}(\chi)) \\ \stackrel{(6.2)}{\Leftrightarrow} & (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models H_{r-i} \wedge \Diamond(H_{r-i+1} \wedge t_{r-i+1}(\chi)) \\ \Leftrightarrow & (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models t_{r-i}(\psi). \end{aligned}$$

Since $\text{depth}_1(\varphi) = r$, $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi$, and $\bar{s} \in V_0 \times W_2$ we get $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{s}) \models t_0(\varphi)$. Hence $t_0(\varphi)$ is interpreted satisfiable in $\mathbf{K4} \times \mathbf{K}$.

For the other direction assume that $t_0(\varphi)$ is interpreted satisfiable in $\mathbf{K4} \times \mathbf{K}$. Thus, there are a transitive structure $\mathfrak{S}_1 = (W_1, \longrightarrow_1, \{W_{1,p} \mid p \in \mathbf{P}_1 \cup \{h_0, \dots, h_r\}\})$ and a structure $\mathfrak{S}_2 = (W_2, \longrightarrow_2, \{W_{2,p} \mid p \in \mathbf{P}_2\})$ and $\bar{s} = \langle s_1, s_2 \rangle \in W_1 \times W_2$ such that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models t_0(\varphi)$. For each $0 \leq i \leq r$ we set

$$T_i = W_{1,h_i} \setminus \left(\bigcup_{j \neq i} W_{1,h_j} \right),$$

corresponding to the formulas H_i in \mathfrak{S}_1 . In particular, we have for all $\bar{x} \in W_1 \times W_2$, $k \in [0, r]$:

$$(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models H_k \quad \Leftrightarrow \quad \bar{x} \in T_k \times W_2. \quad (6.3)$$

Define a structure $\mathfrak{S}'_1 = (W'_1, \longrightarrow'_1, \{W'_{1,p} \mid p \in \mathbf{P}_1\})$ by taking

- $W'_1 = \bigcup_{0 \leq i \leq r} T_i$,
- $\longrightarrow'_1 = \longrightarrow_1 \cap (\bigcup_{0 \leq i < r} T_i \times T_{i+1})$, and
- $W'_{1,p} = W_{1,p} \cap W'_1$ for all $p \in \mathbf{P}_1$.

We prove by structural induction that for each subformula ψ of φ we have: for all $i \in [0, r]$ and all $\bar{x} \in T_{r-i} \times W_2$ it holds

$$\text{depth}_1(\psi) \leq i \quad \Rightarrow \quad ((\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models t_{r-i}(\psi) \quad \Leftrightarrow \quad (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models \psi)$$

For the induction base assume $\psi = p$ for some atomic proposition $p \in \mathbf{P}_1 \cup \mathbf{P}_2$, $\bar{x} \in T_{r-i} \times W_2$, and $i \geq \text{depth}_1(\psi) = 0$ arbitrary. We have $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models t_{r-i}(p)$ iff $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models H_{r-i} \wedge p$ iff $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models p$ where the last equivalence is due to (6.3) and the definition of $W'_{1,p}$.

For the induction step assume that ψ is not atomic, let $i \in [0, r]$ be such that $\text{depth}_1(\psi) \leq i$ and fix an arbitrary $\bar{x} = \langle x_1, x_2 \rangle \in T_{r-i} \times W_2$. We make a case distinction on the structure of ψ . For the cases $\neg\chi$, $\chi_1 \wedge \chi_2$, and $\Diamond\chi$ the equivalence follows directly from the induction hypothesis and (6.3). For the remaining case $\psi = \Diamond\chi$ we have $\text{depth}_1(\chi) = \text{depth}_1(\psi) - 1 \leq i - 1$ and $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models t_{r-i}(\psi)$ is equivalent to:

$$\begin{aligned} & (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models H_{r-i} \wedge \Diamond(H_{r-i+1} \wedge t_{r-i+1}(\chi)) \\ \stackrel{(6.3)}{\Leftrightarrow} & (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{x}) \models \Diamond(H_{r-i+1} \wedge t_{r-i+1}(\chi)) \\ \stackrel{\text{Def. } T_{r-i+1}}{\Leftrightarrow} & \exists \bar{y} \in T_{r-i+1} \times W_2 : \bar{x} \longrightarrow_1 \bar{y} \text{ and } (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{y}) \models H_{r-i+1} \wedge t_{r-i+1}(\chi) \\ \stackrel{(6.3)}{\Leftrightarrow} & \exists \bar{y} \in T_{r-i+1} \times W_2 : \bar{x} \longrightarrow_1 \bar{y} \text{ and } (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{y}) \models t_{r-i+1}(\chi) \\ \stackrel{\text{Def. } \longrightarrow'_1}{\Leftrightarrow} & \exists \bar{y} \in T_{r-i+1} \times W_2 : \bar{x} \longrightarrow'_1 \bar{y} \text{ and } (\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{y}) \models t_{r-i+1}(\chi) \\ \stackrel{IH}{\Leftrightarrow} & \exists \bar{y} \in T_{r-i+1} \times W_2 : \bar{x} \longrightarrow'_1 \bar{y} \text{ and } (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{y}) \models \chi \\ \Leftrightarrow & (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{x}) \models \Diamond\chi \end{aligned}$$

By assumption we have $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models t_0(\varphi)$, $\bar{s} \in T_0 \times W_2$, and $\text{depth}_1(\varphi) = r$. Thus, the above equivalence implies $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi$ and thus, φ is interpreted satisfiable in $\mathbf{K} \times \mathbf{K}$. ■

Lemma 6.11 provides a reduction of $\mathbf{K}_{\text{id}}^2\text{-SAT}$ to interpreted satisfiability in $\mathbf{K4} \times \mathbf{K}$. Similarly, one can give a reduction of $\mathbf{K}_{\text{id}}^2\text{-SAT}$ to interpreted satisfiability in $\mathbf{S4} \times \mathbf{K}$. Finally, Proposition 6.2 together with Theorem 6.8 yields the following result.

Theorem 6.12. *Satisfiability in $\mathbf{K4} \times \mathbf{K}$ and $\mathbf{S4} \times \mathbf{K}$ is nonelementary.*

Next, we study the combination $\mathbf{S5}_2 \times \mathbf{K}$. As already announced, we lift the reduction from $\mathbf{S5}_2$ to \mathbf{K} , see for instance [69, 52], to the two-dimensional case. Intuitively, *one* transition in \mathbf{K} is simulated by *two* transitions in $\mathbf{S5}_2$. This is possible since the composition of two equivalence relations is neither symmetric nor transitive in general and using the fresh variable p^* we can enforce a nontrivial transition, that is we change equivalence classes. Let us define the translation function \cdot^\dagger by

$$\begin{aligned} q^\dagger &= p^* \wedge q && \text{for all } q \in \mathbf{P} \\ (\varphi_1 \wedge \varphi_2)^\dagger &= p^* \wedge \varphi_1^\dagger \wedge \varphi_2^\dagger \\ (\neg \varphi)^\dagger &= p^* \wedge \neg(\varphi^\dagger) \\ (\bar{\Diamond} \varphi)^\dagger &= p^* \wedge \bar{\Diamond} \varphi^\dagger \\ (\Diamond \varphi)^\dagger &= p^* \wedge \Diamond_\equiv (\neg p^* \wedge \Diamond_\approx (p^* \wedge \varphi^\dagger)) \end{aligned}$$

where \Diamond_\equiv and \Diamond_\approx refer to the two modalities in the $\mathbf{S5}_2$ -component and p^* is a fresh propositional variable in the signature of the first component. It is routine to prove the following.

Lemma 6.13. *For every (\mathbf{A}, \mathbf{P}) -formula φ we have: φ is interpreted satisfiable in \mathbf{K}^2 iff φ^\dagger is interpreted satisfiable in $\mathbf{S5}_2 \times \mathbf{K}$.*

Proof. We assume that φ is defined over $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2$ for disjoint \mathbf{P}_1 and \mathbf{P}_2 with $p^* \notin \mathbf{P}_1 \cup \mathbf{P}_2$.

Assume first that φ is interpreted satisfiable in $\mathbf{K} \times \mathbf{K}$. Thus, there are structures $\mathfrak{S}_1 = (W_1, \xrightarrow{a}, \{W_{1,p} \mid p \in \mathbf{P}_1\})$, $\mathfrak{S}_2 = (W_2, \xrightarrow{b}, \{W_{2,p} \mid p \in \mathbf{P}_2\})$, and a world $\bar{s} \in W_1 \times W_2$ such that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi$. Define an $\mathbf{S5}_2$ -structure $\mathfrak{S}'_1 = (W'_1, \equiv, \approx, \{W'_{1,p} \mid p \in \mathbf{P}_1 \cup \{p^*\}\})$ as follows:

- $W'_1 = W_1 \uplus \xrightarrow{a}$,
- \equiv is the reflexive, transitive, and symmetric closure of $\{(w, (w, w')) \mid w \xrightarrow{a} w'\}$,
- \approx is the reflexive, transitive, and symmetric closure of $\{((w, w'), w') \mid w \xrightarrow{a} w'\}$,
- $W'_{1,p} = W_{1,p}$ for $p \in \mathbf{P}_1$,
- $W'_{1,p^*} = W_1$.

Now, one can prove by induction on the structure of a formula ψ that for every world $\bar{w} \in W_1 \times W_2$ we have:

$$(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{w}) \models \psi \quad \Leftrightarrow \quad (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{w}) \models \psi^\dagger.$$

For the induction base, i.e., when ψ is a propositional variable, the statement is immediately true, by definition of the structure \mathfrak{S}'_1 . For the cases $\neg\chi$, $\chi_1 \wedge \chi_2$ and $\overline{\Diamond}\chi$, the statement follows directly from the induction hypothesis.

So assume ψ is of the form $\Diamond\chi$. Suppose first that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models \Diamond\chi$. Thus, there is some world s'_1 such that $s_1 \xrightarrow{a} s'_1$ and $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi$. By induction hypothesis, we have $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi^\dagger$. By definition of \mathfrak{S}'_1 , we have $s_1 \equiv (s_1, s'_1)$, $(s_1, s'_1) \approx s'_1$, $\{s_1, s'_1\} \subseteq W'_{1,p^*}$, and $(s_1, s'_1) \notin W'_{1,p^*}$. Obviously, this yields $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models p^* \wedge \Diamond \equiv (\neg p^* \wedge \Diamond \approx (p^* \wedge \chi^\dagger))$. For the other direction suppose $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models p^* \wedge \Diamond \equiv (\neg p^* \wedge \Diamond \approx (p^* \wedge \chi^\dagger))$. Thus, there are worlds $t, s'_1 \in W'_1$ with $s_1 \equiv t$, $t \approx s'_1$, $s_1, s'_1 \in W'_{1,p^*}$, and $t \notin W'_{1,p^*}$ such that $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi^\dagger$. By definition of \mathfrak{S}'_1 , we know that $s'_1 \in W_1$, $t = (s_1, s'_1)$, and $s_1 \xrightarrow{a} s'_1$. As $s'_1 \in W_1$, the induction hypothesis implies $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi$. Hence, $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models \Diamond\chi$.

In particular, we obtain $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi^\dagger$, and thus, φ^\dagger is interpreted satisfiable in $\mathbf{S5}_2 \times \mathbf{K}$.

Assume now that φ^\dagger is interpreted satisfiable in $\mathbf{S5}_2 \times \mathbf{K}$. Hence, there is an $\mathbf{S5}_2$ -structure

$$\mathfrak{S}_1 = (W_1, \equiv, \approx, \{W_{1,p} \mid p \in P_1 \cup \{p^*\}\}),$$

a structure $\mathfrak{S}_2 = (W_2, \xrightarrow{b}, \{W_{2,p} \mid p \in P_2\})$, and $\bar{s} \in W_1 \times W_2$ such that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi^\dagger$. Define a structure $\mathfrak{S}'_1 = (W'_1, \xrightarrow{a}, \{W'_{1,p} \mid p \in P_1\})$ as follows:

- $W'_1 = W_{1,p^*}$
- $\xrightarrow{a} = \{(u, v) \mid \exists w \in W_1 \setminus W_{1,p^*} : u \equiv w \approx v\}$
- $W'_{1,p} = W_{1,p} \cap W_{1,p^*}$ for all $p \in P_1$

One can prove by induction on the structure of a formula ψ that for every world $\bar{w} \in W'_1 \times W_2$ we have:

$$(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{w}) \models \psi^\dagger \quad \Leftrightarrow \quad (\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{w}) \models \psi$$

Again, the case when ψ is a propositional variable is immediately clear from the definition of \mathfrak{S}'_1 . Also the cases $\neg\chi$, $\chi_1 \wedge \chi_2$, and $\overline{\Diamond}\chi$ are direct consequences of the induction hypothesis.

For the case $\psi = \Diamond\chi$ assume first that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models (\Diamond\chi)^\dagger$. By the semantics, there is some $t \notin W_{1,p^*}$ and $s'_1 \in W_{1,p}$ with $s_1 \equiv t$ and $t \approx s'_1$ such that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi^\dagger$. By induction, we have $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi$. Moreover, the definition of \mathfrak{S}'_1 yields $s_1 \xrightarrow{a} s'_1$. By the semantics, we get $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models \Diamond\chi$. For the other direction assume $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models \Diamond\chi$. Hence, there is some world s'_1 such that $s_1 \xrightarrow{a} s'_1$ and $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi$. By induction, $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s'_1, s_2 \rangle) \models \chi^\dagger$.

By definition of \xrightarrow{a} , there is some $t \in W_1 \setminus W_{1,p^*}$ such that $s_1 \equiv t \approx s'_1$. By definition of \mathfrak{S}'_1 , we have $s_1, s'_1 \in W_{1,p^*}$. Thus, the semantics yields $(\mathfrak{S}_1 \times \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models (\Diamond \chi)^\dagger$.

Observe now that $(\mathfrak{S}_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi^\dagger$ implies $\bar{s} \in W'_1 \times W_2$ by the definition of \dagger . Therefore, we get $(\mathfrak{S}'_1 \times \mathfrak{S}_2, \bar{s}) \models \varphi$ and φ is interpreted satisfiable in $\mathbf{K} \times \mathbf{K}$. ■

The following theorem is an immediate consequence of Lemma 6.13, Proposition 6.8, and Proposition 6.2. It is worth mentioning that the nonelementary lower bound for $\mathbf{S5}_2 \times \mathbf{K}$ is in sharp contrast to NEXPTIME-completeness for satisfiability in $\mathbf{S5} \times \mathbf{K}$ [104].

Theorem 6.14. *Satisfiability in $\mathbf{S5}_2 \times \mathbf{K}$ is nonelementary.*

6.4 Conclusions, open problems, further applications

We have defined a class of trees and a family of formulas enforcing these trees up to bisimulation equivalence. Using these formulas, we were able to show nonelementary lower bounds for satisfiability in $\mathbf{K} \times \mathbf{K}$ and, via reductions from this, for $\mathbf{K4} \times \mathbf{K}$, $\mathbf{S4} \times \mathbf{K}$, and $\mathbf{S5}_2 \times \mathbf{K}$. The applied reductions are lifted from the one-dimensional cases in a straightforward way, so there are potentially more corollaries of this form.

A related interesting open problem is the satisfiability problem for products where one underlying frame is linear. For instance, one can study satisfiability in the logics $\mathcal{L} \times \mathbf{K}$ for $\mathcal{L} \in \{\mathbf{Lin}, \mathbf{Log}(\{(\mathbb{N}, <)\})\}$, $\mathbf{K4.3}$ where:

- **$\mathbf{Lin} \times \mathbf{K}$:** two-dimensional modal logic restricted to products based on frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ where \mathfrak{F}_1 is a frame (W, \xrightarrow{a}) such that \xrightarrow{a} is a linear order on W ;
- **$\mathbf{Log}(\{(\mathbb{N}, <)\}) \times \mathbf{K}$:** two-dimensional modal logic restricted to products based on frames of the form $(\mathbb{N}, <) \times \mathfrak{F}_2$;
- **$\mathbf{K4.3} \times \mathbf{K}$:** two-dimensional modal logic restricted to products based on frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ where \mathfrak{F}_1 is a frame (W, \xrightarrow{a}) and \xrightarrow{a} is transitive and weakly connected.³

In all listed cases, the precise complexity of the satisfiability problem is open and only a nonelementary upper bound is known [52]. We believe that it is not possible to adapt our techniques to these cases, as our technique heavily relies on tree structures; in particular, the enforced trees $\Upsilon_{\ell,n}(j)$ have a very large outdegree of $\text{Tower}(\ell, n)$. Note that for $\mathcal{L} = \mathbf{LTL}$ a nonelementary lower bound is known whose proof, however, uses the ‘until’-operator \mathcal{U} .

One further application of the presented technique, namely showing lower bounds for sizes of *logical decomposition*, is studied in [57]. Full details are out of scope here; however, we want to give some intuition about the implications of our technique. Logical

³A frame (W, \xrightarrow{a}) is *weakly connected* if $s \xrightarrow{a} t$ and $s \xrightarrow{a} t'$ implies $t \xrightarrow{a} t'$ or $t' \xrightarrow{a} t$ for all $s, t, t' \in W$.

decomposition can concisely be summarized as follows: A logic \mathcal{L} admits decomposition with respect to some composition operation op on structures if all \mathcal{L} -properties that are interpreted on structures composed using op , are already determined by the \mathcal{L} -properties of the component structures. As example for a decomposability theorem consider the following special case of a general theorem where \mathcal{L} is modal logic \mathbf{K} and op is the interpreted product.

Theorem 6.15 ([114]). *From an (A, P) -formula φ with $A = A_1 \cup A_2$, $P = P_1 \cup P_2$, one can compute a tuple (Ψ_1, Ψ_2, β) where*

- $\Psi_i = \{\psi_i^j \mid j \in J_i\}$ is a finite set of (A_i, P_i) -formulas for $i \in \{1, 2\}$ and
- β a positive Boolean formula over variables $X = \{x_i^j \mid i \in \{1, 2\}, j \in J_i\}$

such that for every (A_i, P_i) -structure \mathfrak{S}_i and every world s_i of \mathfrak{S}_i :

$$(\mathfrak{S}_1 \times^{\text{id}} \mathfrak{S}_2, \langle s_1, s_2 \rangle) \models \varphi \quad \Leftrightarrow \quad \mu \models \beta.$$

where $\mu : X \rightarrow \{0, 1\}$ is defined by $\mu(x_i^j) = 1$ if and only if $(\mathfrak{S}_i, s_i) \models \psi_i^j$.

This is, for instance, useful in model checking. Intuitively, instead of checking that φ holds in the interpreted product of (\mathfrak{S}_1, s_1) and (\mathfrak{S}_2, s_2) we can perform a number of checks, namely those in Ψ_i , in the components and obtain the result as a Boolean combination β of the results of the checks in the components.

Logical decomposition dates back to the work of Mostowski [106] and Feferman and Vaught [46], where it is shown that first-order logic is decomposable with respect to a general product operation, which covers also disjoint union and product. Later, both for more expressive logics and for more sophisticated operations such decomposability results have been proved, see [103] for a survey. Logical decomposition has several powerful application in computer science. To mention just one example, it should be clear from Theorem 6.15 that data complexity of model checking in \mathbf{K} does not change when going to interpreted product structures: For a fixed formula, also the decomposition is fixed (though possibly large), so data complexity remains the same.

Let us call $D = (\Psi_1, \Psi_2, \beta)$ the *decomposition* of φ and define $|D| = |\beta| + \sum_{i,j} |\psi_i^j|$ to be its *size*. Theorems such as Theorem 6.15 are often proved by showing that the decomposition is effectively computable. However, this typically yields nonelementary sized decompositions. Applying our technique together with techniques from [38], one can show for several cases, that is, logics \mathcal{L} and composition operations op , that this nonelementary blow-up is unavoidable. For instance, one can prove the following theorem.

Theorem 6.16 ([57]). *Every logic that is at least as expressive as and at most elementarily less succinct than modal logic \mathbf{K} does not have elementary sized decompositions in the sense of Theorem 6.15.*

7 Conclusion and Outlook

In this thesis, we have investigated several approaches to enrich classical logics with the goal to improve the modeling of and reasoning about dynamic aspects like uncertainty or change. Common to all approaches considered here is the underlying possible world semantics known from modal logics. The approaches differ, however, in the way the reason over the set of possible worlds. In the first part, probability theory is used to give each world a weight, which intuitively represents the degree of belief that this world is the actual world. We studied two prominent applications in this spirit, namely ontological reasoning in probabilistic first-order logics and ontology-based access to probabilistic data. In the second part, the worlds are viewed as the states of a relational structure and modal logic \mathbf{K} is used both for modeling/reasoning inside the world and for accessing the states in the relational structure. The logic obtained in this combination is called $\mathbf{K} \times \mathbf{K}$. We here restate the central results that were obtained in the thesis. We refrain, however, from mentioning possible future work and open problems since this was treated in detail in the respective chapters.

Ontological reasoning in fragments of probabilistic first-order logics

In Chapters 3 and 4, we revisited probabilistic first-order logic for subjective uncertainty (ProbFO) introduced by Halpern and Bacchus in the early 1990s [67, 13] under computational complexity aspects. For this purpose, we pursued an indepth-study of fragments of ProbFO and provided general and thorough explanations for the computational complexity in these logics. We started with pinpointing the reasons for the disastrous complexity beyond recursive enumerability. Then, we identified a condition, *monodicity*, that in contrast to classical approaches leads to manageable fragments. In a nutshell, monodic ProbFO offers an object-centered account for uncertainty, that is, one can express uncertainty about the properties of individuals but not about the relations among them. We presented a suitable abstraction of probabilistic structures, quasi-models, and showed how to exploit them to show (i) recursive enumerability and axiomatizability of monodic ProbFO, (ii) restrictions to decidable fragments of FO yields decidable fragments of monodic ProbFO. The proof of Point (ii) provides upper bounds for monodic Prob \mathcal{L} for some decidable first-order fragments \mathcal{L} . Most notably, we obtain a matching 2EXPTIME upper bound for monodic ProbGF. Moreover, Point (ii) yields a transparent explanation of the good computational properties of a family of probabilistic description logics (ProbDLs) that were recently introduced [101].

We then turned our attention to members of the mentioned family of ProbDLs. Our particular aim was to investigate whether there are well-behaved variants based on the well-known tractable description logic \mathcal{EL} . We started with monodic fragments of Prob \mathcal{EL} where probabilistic operators are allowed to be applied to concepts only. A bit discouraging, we proved that *any* extension of \mathcal{EL} with a single probabilistic operator $P_{\sim p}$ (only applied to concepts) leads to EXPTIME-hardness of subsumption relative to general TBoxes. However, we were able to show that subsumption relative to *classical* TBoxes is tractable when admitting an arbitrary probabilistic operator $P_{\sim p}C$ with $\sim \in \{>, \geq\}$. When adding probabilistic roles and thus leaving the monodic fragment we were able to provide a PSPACE algorithm for subsumption relative to general TBoxes when probability values are restricted to 0 and 1. This is surprising since so far all two-dimensional extensions of \mathcal{EL} have the same complexity as the corresponding variant of \mathcal{ALC} , which is 2EXPTIME-complete in this case. We also show maximality of this fragment by establishing a 2EXPTIME lower bound for any extension with a single probabilistic operator, applied to concepts or roles.

Ontology-based access to probabilistic data

We argued that in the recently popular setting of ontology-based data access (OBDA) it is sometimes necessary to deal with uncertainty in the data. We addressed this in Chapter 5, by laying out the framework of ontology-based access to probabilistic data (pOBDA) and studying the complexity of query answering therein. On a high level, this framework relates to probabilistic databases in the same way as OBDA relates to classical databases. We were able to exploit this close relation by lifting results and techniques from both the field of OBDA and probabilistic databases to our framework. In particular, we showed the usefulness of first-order rewritings [28] and the dichotomy of UCQs for tuple-independent probabilistic databases [37]. As a first step, we make the observation that a too expressive datamodel leads to intractability of *all* queries. We thus, restrict the datamodel to an open world variant of tuple-independent databases, where each tuple comes with a weight and all tuples are independent from each other. The most important results in this section are:

- a dichotomy for (Boolean, connected) CQ q and DL-Lite TBoxes \mathcal{T} , that is, answering q relative to \mathcal{T} is either possible in polynomial time or #P-hard;
- a concrete characterization of which (q, \mathcal{T}) are tractable;
- every instance query is tractable relative to every DL-Lite TBox, even for more expressive DL-Lite dialects;
- a dichotomy for (connected) CQs q and \mathcal{ELI} TBoxes \mathcal{T} (most notably: non-first-order rewritability leads to #P-hardness).

Hence, similar to probabilistic databases, tractability is an elusive property. We addressed this by also studying approximations of query probabilities via the notion of FPRASes, that is polynomial time approximations. There, the picture is more encouraging. The following are our main results.

- every CQ and DL-Lite TBox admit an FPRAS, even when allowing for a more expressive datamodel than above;
- over this more expressive datamodel, a CQ is tractable relative to an \mathcal{ELI} TBox if, and only if it is first-order rewritable;
- over the standard data model, we show the close connection of non-first-order rewritability to notoriously hard probabilistic reliability problems.

We believe (as argued) that this chapter is most relevant for practical applications, for example for managing automatically extracted information. Moreover, we have shown that first-order rewritability allows us to directly implement the framework on top of existing probabilistic database management systems.

Satisfiability in $\mathbf{K} \times \mathbf{K}$

In the last chapter, we studied the satisfiability problem in the two-dimensional modal logic $\mathbf{K} \times \mathbf{K}$. In particular, we settled the precise complexity for this problem to be complete for nonelementary time by improving the best known NEXPTIME lower bound to a nonelementary lower bound. As a basic preliminary step, and of independent interest, we showed how to enforce nonelementarily branching trees in this logic. These trees were then used to represent arbitrary large elementary numbers. This representation enabled us to reduce from appropriate tiling problems to show k -NEXPTIME-hardness for every $k \geq 1$. In a second step, we lifted the well-known one-dimensional reductions of (satisfiability in) \mathbf{K} to $\mathbf{K4}$, $\mathbf{S4}$, and $\mathbf{S5}_2$ to show hardness for nonelementary time also for the logics $\mathbf{K} \times \mathbf{K4}$, $\mathbf{K} \times \mathbf{S4}$, and $\mathbf{K} \times \mathbf{S5}_2$.

List of Figures

1.1	Example for the possible worlds semantics.	5
1.2	Some data about soccer players and their clubs.	7
1.3	Extracted data in a probabilistic database.	8
2.1	Example for a probabilistic interpretation.	22
3.1	Example for a probabilistic structure.	30
3.2	Idea underlying the Proof of Claim 1.	51
3.3	Axiomatization for monodic ProbFO.	54
3.4	Axiomatization for ordered fields.	55
4.1	Intuition for non-convexity.	78
4.2	TBox completion rules for subsumption in $\text{Prob}\mathcal{EL}_c^{\sim p,=1}$	85
4.3	Illustration of the reasoning in Example 4.15.	96
4.4	Saturation rules for $\text{cl}(\Gamma)$	97
4.5	The rules for completing the data structures.	99
5.1	The rules for computing query probabilities.	129
5.2	Example queries	135
5.3	Gadget for the $\#P$ -hardness proof.	142
6.1	Example of some product structures.	164
6.2	Intuition of trees encoding numbers.	169
6.3	The $(1, 3)$ -tree $\Upsilon_{1,3}(175, \{\mathbf{b}, \min_{\mathbf{b}}\})$	171
6.4	Example grid element tree.	181

Bibliography

- [1] M. Abadi and J. Y. Halpern. Decidability and expressiveness for first-order logics of probability. *Information and Computation*, 112:1–36, 1994.
- [2] H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.
- [3] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *Proceedings of the 26th International Conference on Data Engineering (ICDE 2008)*, pages 983–992, 2008.
- [4] L. Antova, C. Koch, and D. Olteanu. $10^{(10^6)}$ worlds and beyond: efficient representation and processing of incomplete information. *VLDB Journal*, 18(5):1021–1040, 2009.
- [5] A. Artale, R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyashev. Temporalising tractable description logics. In *Proceedings of the Fourteenth International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society Press, 2007.
- [6] A. Artale, C. Lutz, and D. Toman. A description logic of change. In M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 218–223. AAAI Press, 2007.
- [7] E. Artin and O. Schreier. Algebraische Konstruktion reeller Körper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 5:85–99, 1927.
- [8] F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 325–330. Morgan Kaufmann, 2003.
- [9] F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 364–369. Professional Book Center, 2005.
- [10] F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. LTCS-Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005. See <http://lat.inf.tu-dresden.de/research/reports.html>.

- [11] F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope further. In K. Clark and P. F. Patel-Schneider, editors, *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.
- [12] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [13] F. Bacchus. *Representing and reasoning with probabilistic knowledge - a logical approach to probabilities*. MIT Press, 1990.
- [14] F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.
- [15] V. Bárány, B. ten Cate, and L. Segoufin. Guarded negation. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming (ICALP 2011)*, volume 6756, pages 356–367. Springer, 2011.
- [16] M. Bienvenu, C. Lutz, and F. Wolter. Query containment in description logics reconsidered. In *Proceedings of the 13th International Joint Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*. AAAI Press, 2012.
- [17] M. Bienvenu, C. Lutz, and F. Wolter. First-order rewritability of atomic queries in horn description logics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*. IJCAI/AAAI, 2013.
- [18] M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. In *Proceedings of the 32nd ACM Symposium on Principles of Database Systems (PODS 2013)*. ACM Press, 2013.
- [19] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [20] P. Blackburn, F. Wolter, and J. van Benthem, editors. *Handbook of Modal Logic*. Elsevier, 2006.
- [21] P. Bohannon, S. Merugu, C. Yu, V. Agarwal, P. DeRose, A. Iyer, A. Jain, V. Kakade, M. Muralidharan, R. Ramakrishnan, and W. Shen. Purple sox extraction management system. *SIGMOD Record*, 37(4):21–27, Mar. 2008.
- [22] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem*. Universitext. Springer-Verlag, Berlin, 2001.

- [23] J. Boulos, N. N. Dalvi, B. Mandhani, S. Mathur, C. Ré, and D. Suciu. MYSTIQ: a system for finding more answers by using probabilities. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2005)*, pages 891–893, 2005.
- [24] T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera. The satisfiability problem for probabilistic CTL. In *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008)*, pages 391–402. IEEE Computer Society, 2008.
- [25] A. A. Bulatov. The complexity of the counting constraint satisfaction problem. In *Proceedings of 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 646–661. Springer, 2008.
- [26] A. Cali, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2009)*, pages 77–86. ACM, 2009.
- [27] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.
- [28] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.
- [29] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC 1977)*, pages 77–90. ACM, 1977.
- [30] B. S. Chlebus. From domino tilings to a new model of computation. In A. Skowron, editor, *Symposium on Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 24–33. Springer, 1984.
- [31] P. C. G. da Costa and K. B. Laskey. Pr-owl: A framework for probabilistic ontologies. In *Proceedings of the Fourth International Conference on Formal Ontology in Information Systems (FOIS 2006)*, pages 237–249. IOS Press, 2006.
- [32] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: diamonds in the dirt. *Communications of the ACM*, 52(7):86–94, 2009.
- [33] N. N. Dalvi, K. Schnaitter, and D. Suciu. Computing query probability with incidence algebras. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART*

- Symposium on Principles of Database Systems (PODS 2010)*, pages 203–214. ACM, 2010.
- [34] N. N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005)*, pages 805–816. ACM, 2005.
- [35] N. N. Dalvi and D. Suciu. The dichotomy of conjunctive queries on probabilistic structures. In *Proceedings of the 26th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2007)*, pages 293–302. ACM, 2007.
- [36] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB Journal*, 16(4):523–544, 2007.
- [37] N. N. Dalvi and D. Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM*, 59(6):30, 2012.
- [38] A. Dawar, M. Grohe, S. Kreutzer, and N. Schweikardt. Model Theory Makes Formulas Large. In *Proceedings of 34th International Colloquium on Automata, Language, and Programming (ICALP 2007)*, volume 4596. Springer, 2007.
- [39] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 1319–1325. Professional Book Center, 2005.
- [40] R. de Salvo Braz, E. Amir, and D. Roth. A survey of first-order probabilistic models. In *Innovations in Bayesian Networks: Theory and Applications*, pages 289–317. Springer, 2008.
- [41] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. *IEEE Data Engineering Bulletin*, 29(1):64–72, 2006.
- [42] P. Domingos and W. A. Webb. A tractable first-order probabilistic logic. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*. AAAI Press, 2012.
- [43] E. A. Emerson. Temporal and modal logic. In *HANDBOOK OF THEORETICAL COMPUTER SCIENCE*, pages 995–1072. Elsevier, 1995.
- [44] R. Fagin, B. Kimelfeld, and P. G. Kolaitis. Probabilistic data exchange. *Journal of the ACM*, 58(4):15, 2011.

-
- [45] T. Feder and M. Y. Vardi. Monotone monadic snp and constraint satisfaction. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC 1993)*, pages 612–622. ACM, 1993.
 - [46] S. Feferman and R. L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
 - [47] M. Finger, R. Wassermann, and F. G. Cozman. Satisfiability in \mathcal{EL} with sets of probabilistic aboxes. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*. CEUR-WS.org, 2011.
 - [48] J. Flum and M. Grohe. *Parametrized Complexity Theory*. Springer, 2006.
 - [49] N. Fuhr. Probabilistic datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51:2000, 1999.
 - [50] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 15(1):32–66, 1997.
 - [51] T. Furche, G. Gottlob, G. Grasso, O. Gunes, X. Guo, A. Kravchenko, G. Orsi, C. Schallhart, A. J. Sellers, and C. Wang. DIADEM: domain-centric, intelligent, automated data extraction methodology. In *Proceedings of the 21st World Wide Web Conference (WWW 2012)*, pages 267–270. ACM, 2012.
 - [52] D. M. Gabbay, Á. Kurucz, F. Wolter, and M. Zakharyashev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.
 - [53] D. M. Gabbay and V. B. Shehtman. Undecidability of modal and intermediate first-order logics with two individual variables. *Journal of Symbolic Logic*, 58(3):800–823, 1993.
 - [54] D. M. Gabbay and V. B. Shehtman. Products of Modal Logics, Part 1. *Logic Journal of the IGPL*, 6(1):73–146, 1998.
 - [55] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyashev. Products of ‘transitive’ modal logics. *Journal of Symbolic Logic*, 70(3):993–1021, 2005.
 - [56] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.
 - [57] S. Göller, J. C. Jung, and M. Lohrey. The complexity of decomposing modal and first-order theories. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science (LICS 2012)*, pages 325–334. IEEE, 2012.

- [58] S. Göller, J. C. Jung, and M. Lohrey. The complexity of decomposing modal and first-order theories. In *ACM Transactions on Computational Logic*, to appear.
- [59] G. Gottlob, T. Lukasiewicz, and G. I. Simari. Conjunctive query answering in probabilistic datalog+/- ontologies. In *Proceedings of 5th International Conference on Web Reasoning and Rule Systems (RR 2011)*, pages 77–92. Springer, 2011.
- [60] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.
- [61] E. Grädel, Y. Gurevich, and C. Hirsch. The complexity of query reliability. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 227–234. ACM Press, 1998.
- [62] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3:53–69, 1997.
- [63] T. J. Green and V. Tannen. Models for incomplete and probabilistic information. *IEEE Data Engineering Bulletin*, 29(1):17–24, 2006.
- [64] R. Gupta and S. Sarawagi. Creating probabilistic databases from information extraction models. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 965–976. ACM, 2006.
- [65] V. Gutiérrez-Basulto, J. Jung, C. Lutz, and L. Schröder. A closer look at the probabilistic description logic Prob- \mathcal{EL} . In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press, 2011.
- [66] V. Gutiérrez-Basulto, J. Jung, C. Lutz, and L. Schröder. The complexity of probabilistic \mathcal{EL} . In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745 of *CEUR Workshop Proceedings*, 2011.
- [67] J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, 1990.
- [68] J. Y. Halpern. *Reasoning about uncertainty*. MIT Press, 2005.
- [69] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.
- [70] J. Y. Halpern and M. O. Rabin. A logic to reason about likelihood. *Artificial Intelligence*, 32:379–405, 1987.
- [71] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. i. lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

- [72] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.
- [73] A. Herzig. Modal probability, belief, and actions. *Fundamenta Informaticae*, 57(2-4):323–344, 2003.
- [74] R. Hirsch, I. M. Hodkinson, and Á. Kurucz. On modal logics between $K \times K \times K$ and $S5 \times S5 \times S5$. *Journal of Symbolic Logic*, 67(1):221–234, 2002.
- [75] I. Hodkinson. Monodic packed fragment with equality is decidable. *Studia Logica*, 72:185–197, 2002.
- [76] I. Hodkinson. Complexity of monodic guarded fragments over linear and real time. *Annals of Pure and Applied Logic*, 138:94–125, 2006.
- [77] I. Hodkinson, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyashev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *Proceedings of 10th International Symposium on Temporal Representation and Reasoning/4th International Conference on Temporal Logic (TIME-ICTL 2003)*, pages 91–98. IEEE Computer Society, 2003.
- [78] I. Hodkinson, F. Wolter, and M. Zakharyashev. Decidable fragment of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.
- [79] I. Hodkinson, F. Wolter, and M. Zakharyashev. Monodic fragments of first-order temporal logics: 2000-2001 A.D. In *Proceedings of the 8th Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, pages 1–23. Springer, 2001.
- [80] D. E. Holmes and L. C. Jain, editors. *Innovations in Bayesian Networks: Theory and Applications*, volume 156 of *Studies in Computational Intelligence*. Springer, 2008.
- [81] T. Imielinski and W. L. Jr. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984.
- [82] M. Jaeger. Probabilistic reasoning in terminological logics. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR 1994)*, pages 305–316. Morgan Kaufmann, 1994.
- [83] M. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

- [84] J. C. Jung and C. Lutz. Ontology-based access to probabilistic data with owl-ql. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*. Springer, 2012.
- [85] J. C. Jung and C. Lutz. Ontology-based access to probabilistic data. In *Proceedings of the 26th International Workshop on Description Logics (DL 2013)*, pages 258–270. CEUR-WS.org, 2013.
- [86] J. C. Jung, C. Lutz, S. Goncharov, and L. Schröder. Monodic fragments of probabilistic first-order logic. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014)*, 2014.
- [87] D. R. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM Journal on Computing*, 29(2):492–514, 1999.
- [88] R. M. Karp and M. Luby. Monte-carlo algorithms for enumeration and reliability problems. In *Proceedings of 24th Annual Symposium on Foundations of Computer Science (FOCS 1983)*, pages 56–64. IEEE Computer Society, 1983.
- [89] Y. Kazakov. Consequence-driven reasoning for horn shiq ontologies. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 2040–2045. IJCAI/AAAI, 2009.
- [90] S. Kikot, R. Kontchakov, V. V. Podolskii, and M. Zakharyashev. Exponential lower bounds and separation for query rewriting. In *Proceedings of 39th International Colloquium on Automata, Languages and Programming (ICALP 2012)*, pages 263–274. Springer, 2012.
- [91] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.
- [92] D. Koller and J. Halpern. Irrelevance and conditioning in first-order probabilistic logic. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI 1996)*, pages 569–576. AAAI Press/The MIT Press, 1996.
- [93] D. Koller, A. Y. Levy, and A. Pfeffer. P-classic: A tractable probabilistic description logic. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 1997)*, pages 390–397. AAAI Press/The MIT Press, 1997.
- [94] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyashev. The combined approach to query answering in dl-lite. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010.

- [95] H. E. Kyburg. *The Logical Foundation of Statistical Inference*. D. Reidel, Dortrecht, Netherlands, 1974.
- [96] R. E. Ladner. The Computational Complexity of Provability in Systems of Modal Propositional Logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.
- [97] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93, 2002.
- [98] P. S. Laplace. *Essai philosophique sur les probabilités*. Courcier Imprimeur, Paris, 1816.
- [99] T. Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7):852–883, 2008.
- [100] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, 2008.
- [101] C. Lutz and L. Schröder. Probabilistic description logics for subjective uncertainty. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010.
- [102] C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*. AAAI Press, 2012.
- [103] J. A. Makowsky. Algorithmic uses of the Feferman-Vaught Theorem. *Annals of Pure and Applied Logic*, 126(1-3):159–213, 2004.
- [104] M. Marx. Complexity of products of modal logics. *Journal of Logic and Computation*, 9(2):197–214, 1999.
- [105] M. Marx and S. Mikuláš. Products, or How to Create Modal Logics of High Complexity. *Logic Journal of the IGPL*, 9(1):71–82, 2001.
- [106] A. Mostowski. On direct products of theories. *Journal of Symbolic Logic*, 17:1–31, 1952.
- [107] R. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1992.
- [108] R. T. Ng and V. S. Subrahmanian. Stable semantics for probabilistic deductive databases. *Information and Computation*, 110(1):42–83, 1994.
- [109] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 2153–2158. IJCAI/AAAI, 2011.

- [110] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence.*, 28(1):71–87, 1986.
- [111] J. Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.
- [112] D. Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 985–991. Morgan Kaufmann, 2003.
- [113] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.
- [114] A. Rabinovich. On compositionality and its limitations. *ACM Transactions on Computational Logic*, 8(1), 2007.
- [115] L. D. Raedt and K. Kersting. Probabilistic inductive logic programming. In *Proceedings of 15th International Conference on Algorithmic Learning Theory (ALT 2004)*, pages 19–36. Springer, 2004.
- [116] L. D. Raedt, A. Kimmig, and H. Toivonen. Problog: a probabilistic prolog and its application in link discovery. In *In Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2468–2473. AAAI Press, 2007.
- [117] H. Reichenbach. *The Theory of Probability: An Inquiry into the Logical and Mathematical Foundations of the Calculus of Probability*. University of California Press, Berkely and Los Angeles, 1949.
- [118] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.
- [119] B. Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3), 2008.
- [120] A. D. Sarma, O. Benjelloun, A. Y. Halevy, and J. Widom. Working models for uncertain data. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006)*. IEEE Computer Society, 2006.
- [121] M. Schaefer. Complexity of some geometric and topological problems. In *Proceedings of 17th International Symposium on Graph Drawing (GD 2009)*, volume 5849, pages 334–344. Springer, 2010.

- [122] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC 1978)*, pages 216–226. ACM, 1978.
- [123] L. J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, Department of Electrical Engineering, MIT, 1974.
- [124] U. Straccia. Top-k retrieval for ontology mediated access to relational databases. *Information Sciences*, 198:1–23, 2012.
- [125] D. Suciu, D. Olteanu, C. Ré, and C. Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
- [126] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
- [127] M. Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC 1982)*, pages 137–146. ACM, 1982.
- [128] V. V. Vazirani. *Approximation algorithms*. Springer, 2001.
- [129] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proceedings of the 2nd Conference on Innovative Data Systems Research (CIDR 2005)*, pages 262–276, 2005.
- [130] F. Wolter and M. Zakharyashev. Modal description logics: Modalizing roles. *Fundam. Inform.*, 39(4):411–438, 1999.
- [131] F. Wolter and M. Zakharyashev. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied Logic*, 118:133–145, 2002.
- [132] R. Zenklusen and M. Laumanns. High-confidence estimation of small s - t reliabilities in directed acyclic networks. *Networks*, 57(4):376–388, 2011.