Abstracting Probabilistic Models: Relations, Constraints and Beyond

Abstracting Probabilistic Models: Relations, Constraints and Beyond

Vaishak Belle[a,b,*]

[a] *School of Informatics, University of Edinburgh, Edinburgh, UK.*
[b] *Alan Turing Institute, London, UK.*

Abstract

Abstraction is a powerful idea widely used in science, to model, reason and explain the behavior of systems in a more tractable search space, by omitting irrelevant details. While notions of abstraction have matured for deterministic systems, the case for abstracting probabilistic models is not yet fully understood.

In this paper, we provide a semantical framework for analyzing such abstractions from first principles. We develop the framework in a general way, allowing for expressive languages, including logic-based ones that admit relational, deterministic and hierarchical constructs with stochastic primitives. We motivate a definition of consistency between a high-level model and its low-level counterpart, but also treat the case when the high-level model is missing critical information present in the low-level model. We go on to prove properties of abstractions, both at the level of the parameter as well as the structure of the models. We conclude with some observations about how abstractions can be derived automatically.

1. Introduction

Abstraction is a powerful idea widely used in science to explain phenomena at the required granularity. Think of explaining a heart disease in terms of its anatomical components versus its molecular composition. Think of understanding the political dynamics of elections by studying micro level phenomena (say, voter grievances in counties) versus macro level events (e.g., television advertisements, gerrymandering). In particular, in computer science, it is often understood as the process of mapping one representation onto a simpler representation by suppressing irrelevant information. The motivation is three-fold:

(a) When representing complex pieces of knowledge, abstraction can provide a way to structure that knowledge, hierarchically or otherwise, so as to yield descriptive clarity and modularity.
(b) Reasoning over large graphs, programs, and other structures is almost always computationally challenging, and so abstracting the problem domain to a smaller search space is attractive. Even in the case of tractable representations, such as arithmetic circuits [17], reasoning is polynomial in the circuit size, so clearly a smaller circuit is more effective.
(c) Lastly, and perhaps most significantly, abstraction features pervasively in commonsense reasoning, and there is much discussion in the fields of cognitive science and philosophy on the role of abstractions for explanations [41, 18]; for example, Garfinkel [27] argues that concrete explanations containing too much detail are sensitive to perturbations and are impractical for understanding physical phenomena. Thus, abstractions will likely be critical for *explainable AI* [32], and indeed, much of that literature focuses on extracting high-level symbolic and/or programmatic representations from low-level data (e.g., [56, 67]).

Formal perspectives on abstraction have matured considerably over the years [30, 50, 2]. In particular, the work of Banihashemi et al. [2] is noteworthy as it identifies how notions of soundness and completeness relate to the model-theoretic properties of a high-level abstraction and the corresponding low-level theory. However, the formal analysis

*Corresponding author. The author was supported by a Royal Society University Research Fellowship.
Email address: vaishak@ed.ac.uk (Vaishak Belle)

of abstraction has largely focused on categorical (deterministic and non-probabilistic) domains; that is, both the high-level and the low-level representations are assumed to be categorical assertions. In that regard, existing frameworks are not immediately applicable to the fields of probabilistic modeling and statistical machine learning. Indeed, we do not yet have a full understanding of which aspects of one probabilistic model, representing some low-level phenomena, can be omitted when building a less granular (possibly non-probabilistic) model standing for a high-level understanding of the domain.

In this paper, we provide a semantical framework for analyzing such abstractions from first principles. We develop the framework in a general way, allowing for expressive languages, including logic-based ones that admit relational, deterministic and hierarchical constructs with stochastic primitives [35, 29]. Representative examples of such languages include probabilistic databases and statistical knowledge bases, which have received considerable attention both in the academic and industry circles [68, 58, 72, 21, 55, 10].

In this work, we motivate a definition of consistency between a high-level (probabilistic or logical) model and its low-level (probabilistic) counterpart, but also treat the case when the high-level model is missing critical information present in the low-level model. We go on to prove properties of abstractions, both at the level of the parameter as well as the structure of the models. Put differently, we first motivate a definition of abstraction purely at the level of the model theory, which then provides the basis for analyzing the properties of "unweighted abstractions." (That is, probabilities are simply ignored in that construction.) We use that analysis to investigate how "weighted abstractions" can be defined. We then study how to incorporate low-level evidence and reason about it in the high-level representation. We conclude with some observations about how abstractions can be derived automatically.

With the development of this framework, we hope to provide a formal basis for developing probabilistic abstractions in service of increased modularity, tractability and interpretability.

2. Desiderata

Before developing a framework for abstraction, let us briefly reflect on what is desired of such a framework. To a first approximation, a formal theory of abstraction can be approached in three stages:

1. How should abstraction be defined between a high-level representation Δ_h and a low-level one Δ_l?
2. Given Δ_h and Δ_l, how do we prove that Δ_h is an abstraction of Δ_l?
3. Given Δ_l and a target high-level vocabulary, how do we find Δ_h?

At the outset, in this work, we are concerned with (1) and (2), but we will also consider a preliminary investigation of (3).

In essence, abstractions are about omitting irrelevant details, while providing a less granular language to capture and reason about the underlying probabilistic components. To motivate that using an example, consider a probabilistic relational model (PRM) on entity-relationships for a university database $\mathcal{U}$ (adapted from Heckerman et al. [35]). The model instantiates constraints for a (parameterised) Bayesian network:

Difficulty $\longrightarrow$ Grades $\longleftarrow$ IQ

as follows, referred to as the low-level theory $\mathcal{U}_l$ in the sequel:

0.7 $\mathit{diff}(x, E)$

0.1 $\mathit{diff}(x, M)$

0.2 $\mathit{diff}(x, H)$

0.25 $\mathit{iq}(x, L) \wedge \mathit{diff}(y, E) \wedge \mathit{takes}(x, y) \supset \mathit{grades}(x, y, u)$ for $u \in \{7, 8, 9, 10\}$

0.25 $\mathit{iq}(x, L) \wedge \neg\mathit{diff}(y, E) \wedge \mathit{takes}(x, y) \supset \mathit{grades}(x, y, u)$ for $u \in \{5, 6, 7, 8\}$

where the constants E, M, H, L stand for *easy, medium, hard, low* respectively. (A precise encoding will be presented in a subsequent section.)

The first constraint says that for any given course, say B, the probability that its difficulty level is easy is 0.7. The fourth constraint says that for any low IQ student taking an easy course, the probability that his grade is 7 is 0.25, and

likewise, the probability that his grade is 8 is 0.25, and so on. More generally, this theory says that courses come in three levels of difficulty, and when a low IQ student takes an easy course, his grades can be modeled as a uniform distribution on $\{7, 8, 9, 10\}$, and when he does not take an easy course, it is a uniform distribution on $\{5, 6, 7, 8\}$.

A simple yet powerful type of abstraction to apply here is to abstract away the domain. Assuming the above sentences are the only ones of interest to us, we can lump the constants $\{M, H\}$ as N, standing for *not easy*, and lump the mentioned grade values together as $\{5, 6\}, \{7, 8\}, \{9, 10\}$ and denote them as B, O, G, standing for *bad, ok* and *good* respectively. Then, we would obtain the following model, referred to as the high-level theory $\mathcal{U}_h$ in the sequel:[2]

.7 $diff(x, E)$

.3 $diff(x, N)$

.5 $iq(x, L) \wedge diff(y, E) \wedge takes(x, y) \supset grades(x, y, u)$ for $u \in \{O, G\}$

.5 $iq(x, L) \wedge diff(y, N) \wedge takes(x, y) \supset grades(x, y, u)$ for $u \in \{B, O\}$

On closer inspection, the reader may observe that $\mathcal{U}_h$ is, in fact, a very faithful abstraction of $\mathcal{U}_l$, in terms of accurately grouping together probabilistic events. Indeed, we will formally show that the two models agree on a large class of probabilistic queries. The benefit, of course, is that $\mathcal{U}_h$ is defined over a smaller set of random variables.

However, such a faithful alignment may not always be needed, or even feasible. Consider a case where we abstract by grouping definitions and complex formulas using new predicates. Suppose we had a course listing database C. Let C_l be a low-level theory:

.9 $CS(x) \supset diff(x, H)$

.8 $Physics(x) \supset diff(x, E)$

1 $(AI(x) \supset CS(x)) \wedge (Astronomy(x) \supset Physics(x))$

We may want to define a high-level theory C_h that simply uses $Science(x)$ in place of $CS(x)$ and $Physics(x)$. But then the weight on rules such as $Science(x) \supset diff(x, H)$ or $Science(x) \supset diff(x, E)$ may not be immediate to derive, in general. Predicate abstraction can also be used as a strategy to check for probabilistically significant events. For example, an administrator may only be interested in ensuring that all low IQ students enroll in an easy course: $alert \doteq \neg [\forall x, \exists y\ (iq(x, L) \supset (takes(x, y) \wedge diff(y, E)))]$ and specifically, whether that atom ever obtains a non-zero probability. Indeed, the literature on verification and security often approach the reasoning of complex systems by distinguishing *bad* states (e.g., invalid paths, safety conditions) [65], and correspondingly, checking whether such states are probable or improbable. Naturally, by means of a relational language, such definitions can be arbitrarily complex and hierarchical, and different from classical works on categorical abstraction, predicates at every level can denote stochastic primitives.

In that spirit, we show that abstraction can be understood both from the viewpoint of the parameters (i.e., weights and/or probabilities) and structure (i.e., the logical sentences). While we do discuss the case of aligning probabilities exactly between the high-level and low-level models, we also consider the most immediate case of parameter abstraction where one obtains an alignment between the probable and improbable events. When it comes to abstracting structure, we show that one wants to ensure that the high-level model is consistent, and perhaps additionally that it is not missing critical information present at the low-level model. This then motivates a definition of *soundness* and *completeness*.

Our starting point was the work of Banihashemi et al. [2], which introduces a simple way to logically characterize the differences between a high-level theory and the low-level one, via the well-understood notion of isomorphisms. We show how that account can be extended to reason about probabilities by appealing to the formulation of *weighted model counting* [12], which serves as an assembly language for many popular PRMs. The resulting treatment can

[2] Although the abstraction uses the same predicates as $\mathcal{U}_l$, note that some of these are essentially new predicates, with different domains. For example, in $\mathcal{U}_l$, the difficulty ranges over $\{E, M, H\}$ whereas in $\mathcal{U}_h$, it ranges over $\{E, N\}$. The context will make clear whether the predicates and constants are from $\mathcal{U}_l$ or from $\mathcal{U}_h$, and so we do not distinguish symbols from $\mathcal{U}_h$ by means of superscripts and such.

be seen to share much of the simplicity of Banihashemi et al. [2], thereby providing an amenable framework for understanding probabilistic and logical abstractions of PRMs.

We reiterate that our focus here is primarily about the semantic constraints for analyzing abstractions. Thus, at the outset, we assume that we are given a *high-level theory*, capturing the more abstract probabilistic model, and a *low-level theory*, understood as the underlying probabilistic model that is to be abstracted. Nonetheless, we conclude our technical treatment by discussing some ideas for deriving abstractions automatically.

3. Preliminaries

Our technical development will discuss the semantical constraints between different representations, defined in terms of a mapping between probabilistic events. For the purpose of our results, it will be useful to think in terms of these representations being knowledge bases (i.e., sentences in some logical language), over which one defines a measurable space $(S, \mathcal{F}_S)$ [34]. In particular, for any given knowledge base Δ, we imagine S to be some subset of the set of interpretations of Δ. Moreover, when analyzing how precisely two representations agree, we will be considering the probabilities of queries that additionally use logical connectives such as conjunction and negation, and so we will require that measures be well-defined over such connectives.

Concretely, define a relational language *Lang* with predicate symbols of every arity

$$\{P_1(x), \ldots, P_2(x,y), \ldots, P_3(x,y,z), \ldots\},$$

variables $\{x, y, z, \ldots\}$, connectives $\vee, \neg, \wedge, \forall$ and a set of constants $\{c_1, c_2, \ldots\}$, serving as the *domain of discourse* for quantification. To facilitate comparisons between vocabularies, we assume that for each high/low-level theory the relations and domain are finite subsets of this fixed infinite vocabulary. For simplicity, we restrict our attention to probability spaces over finitely many random variables, as would be instantiated from our assumption. This would be applicable to most statistical relational languages, such as probabilistic databases, Markov logic networks and knowledge graphs [68, 58, 21]. Although from a logical viewpoint, we could simply have used a propositional one, we will introduce a relational language, as is usual in the literature [42].[3]

Standard abbreviations apply for connectives: we write $\alpha \supset \beta$ (material implication) to mean $\neg\alpha \vee \beta$, $\alpha \equiv \beta$ (equivalence) to mean $(\alpha \supset \beta) \wedge (\beta \supset \alpha)$, and $\exists x \alpha$ (existential quantification) to mean $\neg\forall x \neg\alpha$. In particular, when the domain is fixed to a finite set D, we write $\forall x\, \alpha(x)$ to mean $\bigwedge_{c \in D} \alpha(c)$. Moreover, $\alpha \wedge \beta$ is equivalent to $\neg(\neg\alpha \vee \neg\beta)$, so in proofs, we only consider the connectives $\{\wedge, \neg\}$.

The set of ground atoms is defined as:

$$\{P(c_1, \ldots, c_k) \mid P \text{ is a relation}, c_i \in D\}.$$

The set of ground literals is obtained from the set of atoms, and their negations. Henceforth, when we write atoms and literals, we will implicitly mean ground ones. We often use p and q to denote atoms, and l and d to denote literals.

A model M is a $\{0, 1\}$ assignment to the set of atoms. Using $\models$ to denote satisfaction, the semantics for a formula ϕ is defined inductively: $M \models p$ for atom p iff $M[p] = 1$; $M \models \neg\phi$ iff $M \models \phi$ does not hold (also written $M \not\models \phi$); $M \models \phi \vee \psi$ iff $M \models \phi$ or $M \models \psi$; and $M \models \phi \wedge \psi$ iff $M \models \phi$ and $M \models \psi$. We write $l \in M$ to mean that $M \models l$ for literal l.

We say a formula ϕ is *satisfiable* iff there is a model M such that $M \models \phi$. We write $\Delta \models \phi$ to mean that in every model M such that $M \models \Delta$, it is also the case that $M \models \phi$. In particular, we say that ϕ is *valid*, written $\models \phi$, iff for every model M, $M \models \phi$.

To prepare for our technical discussion, we discuss some notational conventions. Given a formula Δ, we write $Lang(\Delta)$ to mean the the logical sub-language implicit in Δ: that is, the set of well-formed formulas constructed from relations $\{P_1(x), \ldots\}$ and constants D mentioned in Δ. We can then write $\alpha \in Lang(\Delta)$ to mean such as well-formed formula. Analogously, we write $Lits(\Delta)$ to mean the set of literals obtained from $Lang(\Delta)$. For example, if $\Delta = P(c) \vee Q(c, a)$, then $\neg P(a) \in Lang(\Delta)$, $Q(a, a) \in Lang(\Delta)$, $P(a) \in Lits(\Delta)$, $\neg Q(a, c) \in Lits(\Delta)$, and so on. We often

[3] If there are infinitely many random variables instantiated from the first-order language, we may consider countably additive probability measures [26, 33] or other syntactic conditions, as in, for example, [52, 39, 66, 49, 4].

abuse notation and write $\vec{c} \in D$ to mean that each of the constants mentioned in $\vec{c}$ is taken from D. Finally, given a Δ, when we write $M \models \Delta$, it is implicit here that we take M to be a model for the language $Lang(\Delta)$; that is, it is a $\{0, 1\}$ assignment to the set of atoms in $Lang(\Delta)$. We can make this explicit by writing $M \in Models(Lang(\Delta))$, or simply $M \in Models(\Delta)$ for short.[4]

As hinted above, we will now assume that for any Δ, we are given a measurable space $(S, \mathcal{F}_S)$, where $S \subseteq Models(\Delta)$ [34]. Since $Models(\Delta)$ is finite, let $S = Models(\Delta)$ for simplicity. For this measurable space, we further assume that for every $\alpha \in Lang(\Delta)$, $\Pr(\alpha)$ is a numeric term; that is, every well-defined formula is accorded a probability. We further interpret a conditional probability expression as

$$\Pr(\alpha \mid \beta) = \frac{\Pr(\alpha \wedge \beta)}{\Pr(\beta)}$$

This interpretation places very little restrictions on the computational machinery that one may use [34]. For the sake of concreteness, *weighted model counting* (WMC) [1], for example, is a reasonable fit. We remark that nothing in our technical treatment hinges on using WMC, and we only use the framework to illustrate examples and the encoding for the university PRM. In some cases, we state useful properties of WMC, but these would hold in virtually all statistical relational languages and probabilistic logics [34, 42, 24].

WMC is defined over the models of a propositional formula, and serves as an assembly language for a number of heterogeneous representations, including factor graphs, Bayesian networks, probabilistic databases and probabilistic programs [1, 68, 25]. WMC enjoys a number of interesting properties that makes it particularly well-suited for our endeavor. First, it separates the symbolic representation (i.e., a logical encoding of the probabilistic model) from a weight function denoting the probabilities of variables, which allows us to investigate abstractions both at the level of structures and at the level of parameters. Second, WMC provides a semantic as well as a computational view for probabilistic reasoning. Semantically, the models of propositional formulas map to *states* in probability spaces (i.e., assignments of values to random variables). Computationally, we are able to reuse SAT technology for building exact and approximate solvers [31], while still leveraging context-specific independences [8]. In particular, recent approaches for WMC [12] such as knowledge compilation [17] provide effective ways for enumerating and testing properties on propositional interpretations.

Essentially, WMC extends *model counting*, which is the task of counting the models of a propositional formula [31]. In WMC, weights are additionally accorded to literals, and we are interested in summing the weights of the models, which is then defined in terms of the product of the literal weights. Standard probabilistic inference, WMC and model counting are, in fact, closely related problems, with polynomial time reductions to each other, with their decision versions being #P-hard [1, 70]. Formally,[5]

Definition 1. Suppose Δ is a ground first-order sentence. Suppose w is a function that maps the elements of $Lits(\Delta)$ to $\mathbb{R}^{[0,\infty)}$. Then the WMC of Δ is defined as:

$$\mathrm{WMC}(\Delta, w) = \sum_{M \models \Delta} \prod_{l \in M} w(l)$$

Given a formula $\phi \in Lang(\Delta)$, we can query ϕ wrt evidence e for theory (Δ, w) using:

$$\begin{aligned} \Pr(\phi \mid e, \Delta, w) &= \frac{\mathrm{WMC}(\phi \wedge e \wedge \Delta, w)}{\mathrm{WMC}(e \wedge \Delta, w)} \\ &= \frac{\Pr(\phi \wedge e, \Delta, w)}{\Pr(e, \Delta, w)} \end{aligned} \tag{‡}$$

[4]The reason we go to some length to discuss our notational conventions is this: when we work with a fixed language, the set of relations, literals, and models to consider is immediate. That will no longer be true when we are thinking of different logical languages for high-level and low-level theories, in which case our notation will provide context.

[5]We define WMC at the level of the ground theory. In the literature, however, a special case of WMC is sometimes considered for relational languages, where the weight function maps predicates directly to numbers (e.g., [71]). The intuitive idea is to treat this weight function as a template for all instances of the corresponding predicate, which, on the one hand, simplifies the specification of the weight function, and on the other, admits effective inference. We do not discuss such ideas here as it is orthogonal to the main thrust of this work (cf. penultimate section).

When $e = \mathit{true}$, we simply write $\Pr(\phi, \Delta, w)$. We remark for $\Pr(\phi, \Delta, w)$ to be well-defined, which is assumed, $\mathrm{WMC}(\Delta, w) \neq 0$. (Thus, it is assumed that Δ is satisfiable, and that w does not map all the corresponding literals to 0.) If the context is clear, we often refer to Δ as the *theory*, and to ϕ as the *query* or *event*.

We immediately observe the following property from the definition of WMC.

Theorem 2. *If* $\Delta \models \phi$, *then* $\Pr(\phi, \Delta, w) = 1$. *If* $\Delta \wedge \phi$ *is not satisfiable, then* $\Pr(\phi, \Delta, w) = 0$.

Proof. For the first property, every M such that $M \models \Delta$, $M \models \phi$ also, and so $\mathrm{WMC}(\phi \wedge \Delta, w) = \mathrm{WMC}(\Delta, w)$. For the second, $\mathrm{WMC}(\phi \wedge \Delta, w) = 0$. ∎

Example 3. We illustrate a WMC encoding for $\mathcal{U}_l$ based on the university PRM; the encoding for others considered in this work are analogous. First, note that in atoms such as $\mathit{diff}(x, y)$, the logical variable y captures the possible values of a random variable. Thus, they are to behave like logical functions. Formally, let $\mathcal{U}_l$ be the union of the following, the free variables being implicitly universally quantified from the outside:

- $\mathit{diff}(y, E) \vee \mathit{diff}(y, M) \vee \mathit{diff}(y, H)$
- $f_1(x, y, u) \equiv [\mathit{iq}(x, L) \wedge \mathit{diff}(y, E) \wedge \mathit{takes}(x, y) \supset \mathit{grades}(x, y, u)]$ for $u \in \{7, 8, 9, 10\}$
- $f_2(x, y, u) \equiv [\mathit{iq}(x, L) \wedge \neg\mathit{diff}(y, E) \wedge \mathit{takes}(x, y) \supset \mathit{grades}(x, y, u)]$ for $u \in \{5, 6, 7, 8\}$

The reason we need to introduce auxiliary predicates f_1 and f_2 is because WMC only allows weights on (ground) literals.

We also need the following hard constraints for capturing the logical functions:

$$\exists u(\mathit{diff}(y, u)), \mathit{diff}(y, u) \wedge \mathit{diff}(y, v) \supset u = v$$

$$\exists u(\mathit{grades}(x, y, u)), \mathit{grades}(x, y, u) \wedge \mathit{grades}(x, y, v) \supset u = v$$

Suppose the domain of quantification for the students is only $\{A\}$ and for courses is only $\{B\}$. We then obtain atoms such as:

$$\mathit{diff}(B, E), \mathit{diff}(B, M), \mathit{diff}(B, H), \mathit{iq}(A, L), \mathit{diff}(B, E), \mathit{takes}(A, B), \mathit{grades}(A, B, 7), \ldots$$

with a weight function w_l for positive atoms derived from the parametric specification in an obvious fashion:

$$w_l(\mathit{diff}(B, E)) = .7, \ldots, w_l(f_1(A, B, 7)) = .25, \ldots$$

We let the weight of a negated atom $w_l(\neg a)$ to be $1 - w_l(a)$. Moreover, the ground instances f_1 and f_2 obtain the weights discussed in the parameterized version. The weights of all atoms not mentioning predicates diff, f_1, f_2 is taken to be 1. It then follows that $\Pr(\mathit{diff}(B, E), \mathcal{U}_l, w) = .7$, and $\Pr(\mathit{grades}(A, B, 7) \mid e, \mathcal{U}_l, w) = .25$, where $e = \mathit{takes}(A, B) \wedge \mathit{iq}(A, L) \wedge \mathit{diff}(A, E)$.

4. Abstraction Framework

We assume that the abstraction framework is realized in terms of two types of representations: a *high-level/abstract theory* that is mapped to a pre-existing *low-level/concrete theory*. Essentially, the logical symbols (predicates and constants) may differ arbitrarily between the two theories. In terms of notation, we use the subscript h to refer to components of the high-level theory, and l to refer to that of the low-level theory.

The first step is to formally establish the construct of a *refinement mapping* between the two theories: the mapping associates each high-level *atom* to a low-level *formula*, which may be arbitrarily complex.

Definition 4. Suppose Δ_h and Δ_l are two theories. We say m is a *refinement mapping* from Δ_h to Δ_l iff for all high-level atoms $p \in \mathit{Lang}(\Delta_h)$, $m(p) = \theta_p$ for some $\theta_p \in \mathit{Lang}(\Delta_l)$.[6]

The mapping m is assumed to extend to complex formulas $\phi \in \mathit{Lang}(\Delta_h)$ inductively: for atoms $\phi = p$, $m(\phi)$ is as above; $m(\neg\phi) = \neg m(\phi)$; $m(\phi \wedge \psi) = m(\phi) \wedge m(\psi)$.

[6] When the high-level and low-level theories are defined over the same domain of discourse D, m can have a compact specification of the form $m(P(\vec{x})) = \theta_P(\vec{x})$, where $P(\vec{x})$ is a non-ground predicate, and $\vec{x}$ are the only free variables in θ_P. So effectively the mapping works by substitutions: for every instance $P(\vec{c})$, we have $m(P(\vec{c})) = \theta_P(\vec{c})$, where $\theta_P(\vec{c})$ is obtained from $\theta_P(\vec{x})$ by substituting the free variables $\vec{x}$ by $\vec{c}$.

It is worth noting that a mapping is deliberately asymmetrical in the sense that its range need not include all the atoms of the low-level theory. That is, there may be atoms $q \in Lang(\Delta_l)$, and consequently, also constants and relations, that do not appear in $m(p)$ for every $p \in Lang(\Delta_h)$. After all, abstractions are about omitting irrelevant details.

In general, we will want to use these mappings to discuss model-theoretic properties of the two theories, so we introduce the notion of an isomorphism:

Definition 5. Given a refinement mapping m as above, we say that $M_h \in Models(\Delta_h)$ is m-isomorphic to $M_l \in Models(\Delta_l)$ iff for all atoms $p \in Lang(\Delta_h)$, we have $M_h \models p$ iff $M_l \models m(p)$. We write this as $M_h \sim_m M_l$.

Thus, isomorphism provides a way to align the truth values between high-level atom and low-level formulas. In particular, because of how refinement mappings can be defined for complex formulas, we obtain the following property:

Theorem 6. *Suppose $M_h \sim_m M_l$. Then for all $\phi \in Lang(\Delta_h)$, $M_h \models \phi$ iff $M_l \models m(\phi)$.*

Proof. We prove by induction on ϕ. Base case immediate by definition. Negation: $M_h \models \neg\phi$ iff $M_h \not\models \phi$ iff (by hypothesis) $M_l \not\models m(\phi)$ iff (by semantics) $M_l \models \neg m(\phi)$ iff (by definition) $M_l \models m(\neg\phi)$. Conjunction: $M_h \models \phi \wedge \psi$ iff $M_h \models \phi$ and $M_h \models \psi$ iff (by hypothesis) $M_l \models m(\phi)$ and $M_l \models m(\psi)$ iff (by semantics) $M_l \models m(\phi) \wedge m(\psi)$ iff (by definition) $M_l \models m(\phi \wedge \psi)$. ∎

Example 7. For the university PRM, we provide a mapping $m_{\mathcal{U}}$ below. When free variables appear, we take it to mean that the mapping applies to all substitutions. So, let $m_{\mathcal{U}}$ map $diff(x, E)$, $takes(x, y), iq(x, L)$ from $\mathcal{U}_h$ to the same atoms in $\mathcal{U}_l$, $m_{\mathcal{U}}(diff(x, N)) = diff(x, M) \vee diff(x, H)$, $m_{\mathcal{U}}(grades(x, y, B)) = grades(x, y, 5) \vee grades(x, y, 6)$, $m_{\mathcal{U}}(grades(x, y, O)) = grades(x, y, 7) \vee grades(x, y, 8)$, and $m_{\mathcal{U}}(grades(x, y, G)) = grades(x, y, 9) \vee grades(x, y, 10)$.

Suppose the domain includes a single student A, who takes course B. Suppose M_h is a model of $\mathcal{U}_h$ where $\{iq(A, L), takes(A, B), diff(B, E), grades(A, B, O)\}$ holds. Now consider the model M_l of $\mathcal{U}_l$ where $\{iq(A, L), takes(A, B), diff(B, E), grades(A, B, 7)\}$ holds. It is easy to verify that $M_h \sim_m M_l$, because the main question is whether M_l satisfies $m_{\mathcal{U}}(grades(A, B, O)) = grades(A, B, 7) \vee grades(A, B, 8)$, which it does.

In the following sections, we will discuss the properties of abstractions based on mappings and isomorphisms.

5. Unweighted Abstractions

To obtain intuitions about the properties of abstract models from first principles, we will consider a fundamental type of abstraction: the absence of probabilities.[7] In so much as probabilistic assertions quantify the likelihood of worlds, omitting probabilities still informs us about the possible and the certain, thus allowing us to test whether Δ_h is consistent with Δ_l.

Definition 8. Given a weighted theory (Δ, w), the unweighted setting refers to the case when for all atoms $p \in Lang(\Delta)$, we have $w(p) = w(\neg p) = 1$.

Since probabilities do not occur in the setting, we can establish consistency by checking whether all conclusions by Δ_h (that is, certain events) are also conclusions by Δ_l: in other words, are the conclusions *sound*? We define:

Definition 9. The theory Δ_h is a *sound abstraction* of Δ_l relative to refinement mapping m iff for all $M_l \in Models(\Delta_l)$, there is a $M_h \in Models(\Delta_h)$ such that $M_h \sim_m M_l$.

Theorem 10. *Suppose Δ_h is a sound abstraction of Δ_l relative to m. Then for all $\phi \in Lang(\Delta_h)$:*
(a) if $\Pr(m(\phi), \Delta_l, w_l) > 0$ *then* $\Pr(\phi, \Delta_h, w_h) > 0$*; and (b) if* $\Pr(\phi, \Delta_h, w_h) = 1$ *then* $\Pr(m(\phi), \Delta_l, w_l) = 1$.

[7] Thus, this section can be seen to establish a framework for abstraction in classical (unweighted) model counting.

Proof. For (a), suppose the antecedent holds, which means there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_l \models m(\phi)$. By assumption, there is a $M_h \in \mathit{Models}(\Delta_h)$ such that $M_h \sim_m M_l$, so $M_h \models \phi$, and $\Pr(\phi, \Delta_h, w_h) \neq 0$. (In the unweighted setting, the weight of M_h cannot be 0, since literals cannot get a 0 weight.)

For (b), suppose antecedent, but not consequent. That is only possible when there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_l \not\models m(\phi)$. But by assumption, there must be $M_h \in \mathit{Models}(\Delta_h)$ such that $M_h \sim_m M_l$. So, $M_h \not\models \phi$ by Theorem 6. Thus, $\Pr(\phi, \Delta_h, w_h) \neq 1$. Contradiction. ∎

Example 11. It is easy to check that for the university PRM, $\mathcal{U}_h$ is a sound abstraction of $\mathcal{U}_l$ wrt $m_{\mathcal{U}}$.

It is fairly straightforward to construct trivially unsound abstractions. To see a less obvious example, consider C_l from before, and suppose it also included: $\mathit{CS}(x) \supset \mathit{Programming}(x)$ and $\mathit{Physics}(x) \supset \mathit{Fieldwork}(x)$. And as discussed, let C_h be a high-level theory consisting of the same sentences but with the predicate *Science(x)* used everywhere instead of *CS(x)* and *Physics(x)*.

Suppose B is a *CS*-course. Suppose m_C is a mapping that replaces *Science(x)* by $\mathit{CS}(x) \vee \mathit{Physics}(x)$, but maps every other predicate to itself. Then, we have $\Pr(\phi \mid e, C_h, w_h) = 1$ for $\phi = \mathit{Programming}(B) \wedge \mathit{Fieldwork}(B)$ and $e = \mathit{Science}(B)$, whereas, $\Pr(m(\phi) \mid m(e), C_l, w_l) \neq 1$, because there will be possible worlds where $\mathit{CS}(B) \wedge \neg\mathit{Fieldwork}(B)$.

Sound abstractions ascertain that conclusions by Δ_h are consistent with Δ_l. What about events considered *possible* by Δ_h? Because we are omitting information when constructing an abstract model, it may be that Δ_h entertains an event as possible even though Δ_l does not.

Definition 12. The theory Δ_h is a *complete abstraction* of Δ_l relative to m iff for all $M_h \in \mathit{Models}(\Delta_h)$, there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_h \sim_m M_l$.

Theorem 13. *Suppose Δ_h is a complete abstraction of Δ_l relative to m. Then for all $\phi \in \mathit{Lang}(\Delta_h)$: (a) if $\Pr(\phi, \Delta_h, w_h) > 0$ then $\Pr(m(\phi), \Delta_l, w_l) > 0$; and (b) if $\Pr(m(\phi), \Delta_l, w_l) = 1$ then $\Pr(\phi, \Delta_h, w_h) = 1$.*

Proof. For (a), suppose antecedent. Then there is a $M_h \in \mathit{Models}(\Delta_h)$ such that $M_h \models \phi$. By assumption, there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_h \sim_m M_l$ and so $M_l \models m(\phi)$, and $\Pr(m(\phi), \Delta_l, w_l) \neq 0$.

For (b), suppose antecedent but not consequent. Then, there is a $M_h \in \mathit{Models}(\Delta_h)$ such that $M_h \not\models \phi$. But by assumption, there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_h \sim_m M_l$, and so $M_l \not\models m(\phi)$ by Theorem 6. Thus, $\Pr(m(\phi), \Delta_l, w_l) \neq 1$. Contradiction. ∎

Example 14. The university PRM can be seen as a complete abstraction wrt $m_{\mathcal{U}}$.

To see a case where it is not complete, consider a variant high-level theory $\mathcal{U}'_h$ where we ignore the difficulty of courses and have only one rule: $\mathit{iq}(x, L) \wedge \mathit{takes}(x, y) \supset \mathit{grades}(x, y, u)$ where $u \in \{B, O, G\}$. Suppose the low-level theory is $\mathcal{U}'_l = \mathit{diff}(B, H) \wedge \mathcal{U}_l$, and A is a low-IQ student who takes B. It is easy to see that $\Pr(\phi, \mathcal{U}'_h, w_h) > 0$ for $\phi = \mathit{iq}(A, L) \wedge \mathit{takes}(A, B) \wedge \mathit{grades}(A, B, G)$, because $\mathcal{U}'_h$ says that any of the three grades levels are possible. But clearly, B being a hard course means that $\mathit{diff}(B, H) \wedge m_{\mathcal{U}}(\phi)$ cannot be satisfiable, and so it is a zero-probability event wrt $\mathcal{U}'_l$.

Definition 15. The theory Δ_h is a *sound and complete abstraction* of Δ_l relative to m iff Δ_h is both a sound and a complete abstraction of Δ_l relative to m.

Theorem 16. *Suppose Δ_h is a sound and complete abstraction of Δ_l relative to m. Then for every $\phi \in \mathit{Lang}(\Delta_h)$, (a) $\Pr(\phi, \Delta_h, w_h) > 0$ iff $\Pr(m(\phi), \Delta_l, w_l) > 0$; and (b) $\Pr(\phi, \Delta_h, w_h) = 1$ iff $\Pr(m(\phi), \Delta_l, w_l) = 1$.*

Proof. Follows from Theorems 10 and 13. ∎

6. Weighted Abstractions

Clearly the above theorems would not hold in general when considering non-trivial weights. It is easy to imagine a weight function that redistributes weights such that zero probability events in Δ_l have high probabilities in Δ_h, and vice versa. So, outside the case of probabilities mapping exactly between Δ_h and Δ_l (discussed in the next section), we need to understand how to abstract weighted theories. The previous section provided a recipe for abstractions, from which properties discussed in Theorems 10 and 13 followed. To a first approximation, then, we can motivate a definition for

weighted abstractions by requiring that those properties hold categorically, in the form of *constraints*. But it turns out, we can do better. We can show that if the property about probable events hold as a constraint wrt a sound or complete abstraction, then the corresponding property about certain events follows as a consequence. (Recall that this duality is not about an event and its negation, which would follow from the axioms of probability, but about how the high-level and low-level theories align.)

To prepare for this approach, let us begin with a few properties that follow from the axioms of probability [24], but are established here using WMC:

Theorem 17. *Suppose* (Δ, w) *is a weighted theory. Then the following hold for all* $\phi, \psi \in \mathit{Lang}(\Delta)$*:*

1. *If* $\Delta \models \phi$ *then* $\Pr(\phi, \Delta, w) = 1$.
2. *If* $\phi \wedge \Delta$ *is not satisfiable, then* $\Pr(\phi, \Delta, w) = 0$.
3. $\Pr(\neg\phi, \Delta, w) = 1 - \Pr(\phi, \Delta, w)$.
4. $\Pr(\phi \vee \psi, \Delta, w) = \Pr(\phi, \Delta, w) + \Pr(\psi, \Delta, w) - \Pr(\phi \wedge \psi, \Delta, w)$.
5. *If* $\Pr(\phi, \Delta, w) = 0$ *then* $\Pr(\phi \wedge \psi, \Delta, w) = 0$.
6. *If* $\Pr(\phi, \Delta, w) > 0$ *then* $\Pr(\phi \vee \psi, \Delta, w) > 0$.
7. $\Pr(\phi, \Delta, w) \geq \Pr(\phi \wedge \psi, \Delta, w)$.

Proof. Proofs for (1) and (2) are already discussed in Theorem 2. For (3), we use the fact that $\mathit{Models}(\Delta) = \mathit{Models}(\Delta \wedge \phi) \cup \mathit{Models}(\Delta \wedge \neg\phi)$, and $|\mathit{Models}(\Delta)| = |\mathit{Models}(\Delta \wedge \phi)| + |\mathit{Models}(\Delta \wedge \neg\phi)|$. For (4), we use $\mathit{Models}((\phi \vee \psi) \wedge \Delta) = \mathit{Models}(\phi \wedge \Delta) \cup \mathit{Models}(\psi \wedge \Delta)$ but $|\mathit{Models}((\phi \vee \psi) \wedge \Delta)| = |\mathit{Models}(\phi \wedge \Delta)| + |\mathit{Models}(\psi \wedge \Delta)| - |\mathit{Models}(\psi \wedge \phi \wedge \Delta)|$. For (5), we see that $\Delta \wedge \phi$ has no model (or only zero weight models), and so that clearly also holds for $\Delta \wedge \phi \wedge \psi$. For (6), the models for $\Delta \wedge \phi$ yield a non-zero probability, and these are clearly included in the models for $\Delta \wedge (\phi \vee \psi)$. For (7), the models of $\phi \wedge \psi$ must be a subset (not necessarily proper) of the models of ϕ. ∎

Definition 18. The theory (Δ_h, w_h) is a *weighted sound abstraction* of (Δ_l, w_l) relative to refinement mapping m iff Δ_h is a sound abstraction of Δ_l relative to m, and for all $d \in \mathit{Lits}(\Delta_h)$, if $\Pr(m(d), \Delta_l, w_l) > 0$ then $\Pr(d, \Delta_h, w_h) > 0$.

We will now show that this stipulation at the level of literals immediately implies the validity of the constraint for all formulas:

Theorem 19. *Suppose* (Δ_h, w_h) *is a weighted sound abstraction of* (Δ_l, w_l) *relative to* m*. Then for all* $\phi \in \mathit{Lang}(\Delta_h)$*, if* $\Pr(m(\phi), \Delta_l, w_l) > 0$ *then* $\Pr(\phi, \Delta_h, w_h) > 0$.

Proof. By induction on ϕ. The case of atoms and negations is immediate by definition. So we only need an argument for disjunctions. Suppose $\Pr(m(\phi \vee \psi), \Delta_l, w_l) > 0$, that is, by definition, $\Pr(m(\phi) \vee m(\psi), \Delta_l, w_l) > 0$. By Theorem 17 (4), $\Pr(m(\phi), \Delta_l, w_l) + \Pr(m(\psi), \Delta_l, w_l) - \Pr(m(\phi) \wedge m(\psi), \Delta_l, w_l) > 0$. This is of the form $x + y - z > 0$, where $x, y, z \geq 0$ since these are probabilities. We have 3 cases.

Case $x = 0$: We note that $z = 0$ too, by Theorem 17 (5). So $y > 0$. By hypothesis, $\Pr(\psi, \Delta_h, w_h) > 0$, and therefore $\Pr(\phi \vee \psi, \Delta_h, w_h) > 0$ by Theorem 17 (6).

Case $y = 0$: Symmetric to $x = 0$.

Case $x \neq 0$ and $y \neq 0$: By hypothesis, $\Pr(\phi, \Delta_h, w_h) > 0$ and $\Pr(\psi, \Delta_h, w_h) > 0$. Even if $\Pr(\phi \wedge \psi, \Delta_h, w_h) > 0$, by Theorem 17 (7), it must be that it is smaller or equal to the other probabilities. (That is, if $a, b, c > 0$, $c \leq a$ and $c \leq b$, then $a + b - c > 0$.) So, $\Pr(\phi \vee \psi, \Delta_h, w_h) > 0$. ∎

The key result of this definition is that the property on certain events, seen in Theorem 10 follows as a consequence:

Theorem 20. *Suppose* (Δ_h, w_h) *is a weighted sound abstraction of* (Δ_l, w_l) *relative to* m*. Then for all* $\phi \in \mathit{Lang}(\Delta_h)$*, if* $\Pr(\phi, \Delta_h, w_h) = 1$ *then* $\Pr(m(\phi), \Delta_l, w_l) = 1$.

Proof. Suppose antecedent but not consequent. Then there is some $M_l \in \mathit{Models}(\Delta_l)$ such that $M_l \not\models m(\phi)$ and it has non-zero weight. (If all such M_l have zero weight, then the consequent cannot be falsified because these models do not influence the probability.) By assumption, there is a $M_h \in \mathit{Models}(\Delta_h)$ such that $M_h \sim_m M_l$, and so $M_h \not\models \phi$.

There are now two cases, depending on the weight of the model M_h. (And so the proof deviates from that for Theorem 10.)

Case $w_h(M_h) \neq 0$: The proof follows as in Theorem 10, yielding a contradiction.

Case $w_h(M_h) = 0$: Let $M_h^{\downarrow}$ be a formula denoting the conjunction of the literals true at M_h. (Since there are finitely many atoms, such a formula can be obtained.) Because $M_h \sim_m M_l$, $M_l \models m(M_h^{\downarrow})$. Overloading the notation $M^{\downarrow}$ to mean conjunction and set of literals true at M, $m(M_h^{\downarrow}) \subseteq M_l^{\downarrow}$, the latter being the set of literals true at M_l. But by assumption M_l has non-zero weight, which means $\Pr(M_l^{\downarrow}, \Delta_l, w_l) > 0$. It follows that $\Pr(m(M_h^{\downarrow}), \Delta_l, w_l) > 0$, because otherwise Theorem 17 (5) would be contradicted. By Theorem 19, $\Pr(M_h^{\downarrow}, \Delta_h, w_h) > 0$, and so $w_h(M_h) \neq 0$. Contradiction. ∎

Example 21. The university PRM can be seen to be a weighted sound abstraction wrt $m_{\mathcal{U}}$.

Consider the university PRM with a variant high-level theory $\mathcal{U}_h''$, where the third constraint is the following instead:

1 $iq(x, L) \wedge diff(y, E) \wedge takes(x, y) \supset grades(x, y, G)$

Consider the query $\phi = iq(A, L) \wedge diff(B, E) \wedge takes(A, B) \supset grades(A, B, O)$. Clearly, the low-level theory accords a non-zero probability to $m_{\mathcal{U}}(\phi)$, but because of the third constraint, $\mathcal{U}_h''$ accords a zero probability to ϕ. Thus, this is not a sound weighted abstraction.

Following these results, extending complete abstractions as well as sound and complete abstractions is analogous, which we state here for the sake of completeness. (The proofs are also analogous and hence omitted.)

Definition 22. The theory (Δ_h, w_h) is a *weighted complete abstraction* of (Δ_l, w_l) relative to refinement mapping m iff Δ_h is a complete abstraction of Δ_l relative to m, and for all $d \in Lits(\Delta_h)$, if $\Pr(d, \Delta_h, w_h) > 0$ then $\Pr(m(d), \Delta_l, w_l) > 0$.

Theorem 23. *Suppose* (Δ_h, w_h) *is a weighted complete abstraction of* (Δ_l, w_l) *relative to* m*. Then for all* $\phi \in Lang(\Delta_h)$*, (a) if* $\Pr(m(\phi), \Delta_l, w_l) = 1$ *then* $\Pr(\phi, \Delta_h, w_h) = 1$*; and (b) if* $\Pr(\phi, \Delta_h, w_h) > 0$ *then* $\Pr(m(\phi), \Delta_h, w_h) > 0$.

Example 24. The university PRM can be seen to be a weighted complete abstraction wrt $m_{\mathcal{U}}$.

Example 14 also applies as an instance of an abstraction that is not weighted complete via:

.33 $iq(x, L) \wedge takes(x, y) \supset grades(x, y, u)$ where $u \in \{B, O, G\}$.

Mainly because the difficulty of courses is ignored, an event is considered probable by the high-level theory but not by the low-level one.

Definition 25. The theory (Δ_h, w_h) is a *weighted sound and complete abstraction* of (Δ_l, w_l) relative to refinement mapping m iff it is both a weighted sound and a weighted complete abstraction.

Theorem 26. *Suppose* (Δ_h, w_h) *is a weighted sound and complete abstraction of* (Δ_l, w_l) *relative to* m*. Then for all* $\phi \in Lang(\Delta_h)$*, (a)* $\Pr(m(\phi), \Delta_l, w_l) = 1$ *iff* $\Pr(\phi, \Delta_h, w_h) = 1$*; and (b)* $\Pr(m(\phi), \Delta_l, w_l) > 0$ *iff* $\Pr(\phi, \Delta_h, w_h) > 0$.

Proof. Follows as a corollary from the results on weighted sound, and weighted complete abstractions. ∎

7. Exact Abstractions

The most faithful case of aligning the high-level and low-level theories is when the probabilities coincide for all high-level queries.[8]

Definition 27. The theory (Δ_h, w_h) is a *weighted exact abstraction* of (Δ_l, w_l) relative to refinement mapping m iff Δ_h is a sound and complete abstraction of Δ_l relative to m, and for all $\phi \in Lang(\Delta_h)$, $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$.

Example 28. The university PRM can be seen to be an instance of a weighted exact abstraction wrt $m_{\mathcal{U}}$. (Below we will consider instances that are not weighted exact abstractions by first appealing to variant definition.)

[8]The distribution on the high-level theory is essentially a "push-forward" measure [69].

Definition 27 naturally generalizes previous definitions on weighted and unweighted abstractions in terms of first stipulating that the logical representations align, and then insisting that probabilities match. One might wonder, of course, why stipulate the former, and not simply rely on the latter, which would make the treatment much simpler. In particular, we put forward a definition of *weak exact abstractions*:

Definition 29. The theory (Δ_h, w_h) is a *weak exact abstraction* of (Δ_l, w_l) relative to refinement mapping m iff for all $\phi \in Lang(\Delta_h)$, $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$.

It is immediate to see that since the constraint for weak exact abstraction is embedded in the notion of a weighted exact abstraction, if a high-level theory is a weighted exact abstraction then it also a weak exact abstraction.

Proposition 30. *Suppose (Δ_h, w_h) and (Δ_l, w_l) are theories and m is a refinement mapping. Suppose (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l). Then (Δ_h, w_h) is a weak exact abstraction of (Δ_l, w_l).*

Proof. By assumption, (a) Δ_h is a sound and complete abstraction of Δ_l relative to m, and (b) for all $\phi \in Lang(\Delta_h)$, $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$. Because of (b), the claim is immediate. ∎

Needless to say, weak exact abstractions do not imply weighted exact abstractions, as there is no requirement that isomorphisms hold between the models of Δ_h and Δ_l. Formally:

Theorem 31. *Suppose (Δ_h, w_h) and (Δ_l, w_l) are theories and m is a refinement mapping. Suppose (Δ_h, w_h) is a weak exact abstraction of (Δ_l, w_l) relative to m. Then it does not follow that (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l).*

To prove this claim, it suffices to provide an example that is a weak exact abstraction but not a weighted exact abstraction. So any unsound and/or incomplete abstraction where the probabilities of high-level atoms are made to match the low-level mappings would do. Let us consider a simple instance based on Example 11.

Example 32. Based on Example 11, consider a low-level theory C'_l with only two sentences *CS(x)* $\supset$ *Programming(x)* and *Physics(x)* $\supset$ *Fieldwork(x)*. Let us consider a high-level theory C'_h with the sentences *Science(x)* $\supset$ *Programming(x)* and *Science(x)* $\supset$ *Fieldwork(x)*. Thus, we are considering a mapping m that maps *Science(x)* to *CS(x)* $\vee$ *Physics(x)*, and *Programming(x)* and *Fieldwork(x)* to themselves. As discussed for Example 11, given a constant B, this abstraction is unsound because there is a low-level model M_l where say $CS(B) \wedge Programming(B) \wedge \neg Physics(B) \wedge \neg Fieldwork(B)$ holds, but there cannot be any model M_h of C'_h where $Science(B) \wedge Programming(B) \wedge \neg Fieldwork(B)$ is true.

Let us now assign probabilities as follows: every world where

$$(CS(x) \vee Physics(x)) \wedge \neg Programming(x) \wedge \neg Fieldwork(x)$$

is falsified obtains a weight of 0. It is now easy to see that *Science(x)* $\supset$ *(Programming(x)* $\wedge$ *Fieldwork(x))* at the high-level, and correspondingly *(CS(x)* $\vee$ *Physics(x))* $\supset$ *(Programming(x)* $\wedge$ *Fieldwork(x))* at the low-level, are effectively hard constraints. It is now also easy to distribute weights such that the probabilities align between the two theories, and therefore we would get a weak exact abstraction but not a weighted exact abstraction.

Here is a particularly extreme case involving an inconsistency:

Example 33. Let Δ_l be any theory and let $\Delta_h = \Delta_l \wedge (p \vee q)$, where $\{p, q\}$ are fresh propositions not present in Δ_l. Let w_l be any weight function for Δ_l which maps atoms r in Δ_l to a number in $[0, 1]$, with the understanding that $w_l(\neg r) = 1 - w_l(r)$. Let s be any atom in Δ_l, and so s is an atom in Δ_h by construction. Let m be a mapping that maps every atom in Δ_l to itself, and $m(p) = s \vee \neg s$ and $m(q) = s \wedge \neg s$. Furthermore, for every atom $r \in \Delta_h - \{p, q\}$, let $w_h(r) = w_l(r)$, $w_h(p) = 1$ (so, $w_h(\neg p) = 0$) and $w_h(q) = 0$ (so, $w_h(\neg q) = 1$). That is, $\Pr(p, \Delta_h, w_h) = 1 = \Pr(m(p), \Delta_l, w_l), \Pr(q, \Delta_h, w_h) = 0 = \Pr(m(q), \Delta_l, w_l)$, and the probabilities of all others atoms in Δ_l are the same in Δ_h. It is thus clear that (Δ_h, w_h) is a weak exact abstraction of (Δ_l, w_l). In particular, for any $M_h \in Models(\Delta_h)$ such that $M_h \models q$, we have $w_h(M_h) = 0$ and by extension, letting $M_h^{\downarrow}$ be the formula denoting the conjunction of literals that are true at M_h, $\Pr(M_h^{\downarrow}, \Delta_h, w_h) = 0$. It follows also that $\Pr(m(M_h^{\downarrow}), \Delta_l, w_l) = 0$. Analogously, for all models $M_h \in Models(\Delta_h)$ such that $M_h \not\models q$, by construction, we have $\Pr(M_h^{\downarrow}, \Delta_h, w_h) = \Pr(m(M_h^{\downarrow}), \Delta_l, w_l)$.

However, by construction, there is a $M_h \in Models(\Delta_h)$ (for example, one where $M_h \models q$) such that there is no $M_l \in Models(\Delta_l)$ where $M_l \models m(q)$. So Δ_h is not a sound and complete abstraction of Δ_l relative to m.

Thus, weighted exact abstractions is a stronger requirement than weak exact abstractions. Clearly, weak exact abstractions would be much more attractive as they involve fewer checks than weighted exact abstractions, the former only requiring probabilistic alignment whilst the latter also insisting on logical alignment. So what is to be gained by the stronger requirement? The answer may depend on the application context. The stronger requirement guarantees downward compatibility with results like Theorems 16 and 26, and so we can be assured about the correctness of the abstraction at the qualitative level. For example, when there is only partial knowledge about probabilities, one may express this knowledge in the form of constraints (e.g., the probability of event α is ≥ 0.4) [33, 24, 6]. In this case, establishing logical alignment may be worthwhile in the first instance, either until that partial knowledge is resolved, or in addition to abstracting such constraints. Analogously, probabilities in most real-world applications are typically learnt from data in a parameter estimation step [43]. In this case, either because the data is not complete or because observations are obtained at run-time in an online setting, the posteriors might change and so constructing weak exact abstractions may not be worthwhile. Indeed, if we expect the parameters of the low-level theory to change, we could consider unweighted abstractions so as to construct a high-level theory that is not sensitive to weights. We would then match probabilities for a particular training epoch with the added assurance that at least the syntactic form of the high-level theory does not change from epoch to epoch.

8. Abstracting Evidence

Recall that we can query ϕ wrt evidence e for theory (Δ, w) using :

$$\Pr(\phi \mid e, \Delta, w) = \frac{\text{WMC}(\phi \wedge e \wedge \Delta, w)}{\text{WMC}(e \wedge \Delta, w)} = \frac{\Pr(\phi \wedge e, \Delta, w)}{\Pr(e, \Delta, w)}$$

We assumed so far that $\phi, e \in \mathit{Lang}(\Delta)$. However, in many applications needing abstraction, it is often the case that observations are low-level (e.g., readings on sensor), whereas the query is at the high-level (e.g., interactions with user). In this section, we discuss some ways to reconcile this issue.[9]

Consider low-level evidence $e \in \mathit{Lits}(\Delta_l)$. For simplicity, let e be a literal. Without loss of generality, let mappings be in conjunctive normal form (CNF). We say a literal is *pure* in a CNF θ if its complement does not appear in θ. (E.g., p is pure in $p \vee q$ but not in $\neg p \vee q$; in contrast, $\neg p$ is pure in $\neg p \vee q$ but not in $p \vee q$.) We observe that, by construction, there may be many high-level atoms that map to formulas involving e. So, given a mapping m, let us retrieve these by *concretization*:

$$m^{-1}(e) = \{\text{atom } p \in \mathit{Lang}(\Delta_h) \mid e \text{ is mentioned \& pure in } m(p)\}.$$

(That is, $m(p)$ is a CNF formula.) Here, $m^{-1}(e)$ is equivalently expressed as a formula: $\bigvee p_i$. The idea is that by looking at high-level atoms where e is pure under the mapping, we are essentially finding atoms that agree with the evidence (and not its negation).

We can now retrieve all low-level sentences these map to by re-applying m as follows: $m(m^{-1}(e)) = \bigvee m(p_i)$. (It is easy to see that e will remain pure in $m(m^{-1}(e))$.)

An immediate case, then, of conditioning being straightforward is when $e = m(m^{-1}(e))$:

Theorem 34. *Suppose (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l) relative to m. Suppose $e \in \mathit{Lits}(\Delta_l)$ and $e = m(m^{-1}(e))$. Then for any $\phi \in \mathit{Lang}(\Delta_h)$, $\Pr(\phi \mid m^{-1}(e), \Delta_h, w_h) = \Pr(m(\phi) \mid e, \Delta_l, w_l)$.*

Proof. By assumption, the probability of $\phi \wedge m^{-1}(e)$ wrt Δ_h must be the same as that of $m(\phi) \wedge m(m^{-1}(e))$ at the low-level. ∎

A simple example is the case of $\mathit{diff}(x, E)$ in the university PRM, as it was mapped to the same atom at both levels.

But beyond this simple case, it is not always possible to reason about low-level events in an exact manner at the high-level. Indeed, as mentioned before, omitting details is the very goal of abstraction. For example, in the university PRM, given any course B, $\Pr(\mathit{diff}(B, M), \mathcal{U}_l, w_l) = .1$, but clearly there is no way to syntactically arrange

[9] It is conceivable that there may be other approaches for this reconciliation, and in our inquiry as well, it will become clear that a number of variants present themselves. We also limit the discussion to weighted exact abstractions for simplicity.

$\{diff(B, E), diff(B, N)\}$ in $\mathcal{U}_h$ to obtain that number. Of course, it would not be hard to show a more involved property, such as $\Pr(diff(B, N), \mathcal{U}_h, w_h) \geq \Pr(diff(B, M), \mathcal{U}_l, w_l)$.

Rather than treating such properties, we will consider the case where probabilities can correspond exactly. Then, one way to incorporate low-level evidence is to *weaken* it, in the sense that conditioning wrt the low-level theory would suffer from a loss in detail, which is precisely the problem faced by the high-level theory. We may think of using $m(m^{-1}(e))$, for example. However, that is not sufficient for conditioning to be *correct*, because $m(m^{-1}(e))$ can say *more* and *less* than e. For example, in the university PRM, suppose we have evidence $e = diff(B, M)$ for $\mathcal{U}_l$. So $m_{\mathcal{U}}^{-1}(e) = diff(B, N)$, and $m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e)) = diff(B, M) \vee diff(B, H)$, which is saying less than e. This is reasonable. But suppose for the sake of the argument, $m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e)) = (diff(B, M) \vee diff(B, H)) \wedge \neg iq(B, L)$. This is somewhat artificial but well-defined: that is, the mapping seems to correlate the fact that B takes a medium or hard course with the fact that the student does not have a low IQ. The problem is that e does not *imply* anything about the IQ of B. Thus, if we use $m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e))$ as evidence, we will be falsely assuming facts that were not observed. We need to ensure that $m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e))$ may say as much or even less than e, but never more, which, as argued, is tantamount to assuming facts that are not observed.

To get around this, we stipulate this implication formally:

Definition 35. Given evidence e and mapping m, we define the m-weakening of e as $m(m^{-1}(e))$. We say this weakening is *deterministic* when $e \models m(m^{-1}(e))$.

As illustrated above, it may not always be the case that $e \models m(m^{-1}(e))$. Perhaps the most obvious (and reasonable) case where determinism follows is when $m(m^{-1}(e))$ is a clause, that is, a disjunction of literals. Because e is pure in $m(m^{-1}(e))$, it immediately follows that $e \models m(m^{-1}(e))$. (E.g., p is pure in $p \vee q$, and of course $p \models p \vee q$.)

Theorem 36. *Suppose (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l) relative to m. Suppose $e \in Lits(\Delta_l)$ and its m-weakening is deterministic. Then,* $\Pr(\phi \mid m^{-1}(e), \Delta_h, w_h) = \Pr(m(\phi) \mid m(m^{-1}(e)), \Delta_l, w_l)$.

Proof. Proof analogous to Theorem 34. ∎

Example 37. For the university PRM and $e = diff(B, M)$, its $m_{\mathcal{U}}$-weakening is $diff(B, M) \vee diff(B, H)$. And indeed, $e \models m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e))$. For the query $\phi = iq(A, L) \wedge takes(A, B) \wedge grades(A, B, O)$, its probability given $m_{\mathcal{U}}^{-1}(e) = diff(B, N)$ at the high-level coincides with the probability of $iq(A, L) \wedge takes(A, B) \wedge (grades(A, B, 7) \vee grades(A, B, 8))$ given $m_{\mathcal{U}}(m_{\mathcal{U}}^{-1}(e))$ at the low-level.

It is also worthwhile to consider the case where $e \equiv m(m^{-1}(e))$. This is the situation where, say, e has just as much information at the low-level as $m^{-1}(e)$ would at the high-level. In the university PRM, for example, if all we observe is $(diff(B, M) \vee diff(B, H))$ at the low-level, this is precisely as informative as observing $diff(B, N)$ at the high-level in the context of a map $m(diff(B, N)) = (diff(B, M) \vee diff(B, H))$. Of course, since $e \equiv m(m^{-1}(e))$ immediately implies $e \models m(m^{-1}(e))$, the above theorem holds as a consequence.

Equivalently, we could consider the straightforward case where the observation is, in fact, at the high-level. That is, we observe e' which is an atom at the high-level, and in the setting of weighted exact abstractions, by way of Definition 27, we immediately get that for all $\phi \in Lang(\Delta_h)$, $\Pr(\phi \mid e', \Delta_h, w_h) = \Pr(m(\phi) \mid m(e'), \Delta_l, w_l)$.

Note, in particular, that when m maps a predicate at the high-level to the same predicate at the low-level, then, of course, low-level evidence $e = m^{-1}(e) = m(m^{-1}(e))$, which is a degenerate case where the above notions collapse.

9. Deriving Abstractions

The main thrust of this paper is on the semantical properties of abstractions, formulated under the assumption that we are given the high-level and low-level theory and the appropriate refinement mapping. Based on these properties, we will now motivate a few directions for deriving abstractions automatically. These directions are to be seen as schemas that appeal to exhaustive search, and so are not necessarily efficient. In particular, they identify general properties that hold, based on which special tractable cases, or variations, may be considered.

9.1. Formula substitutions

Rather than deploying a general search procedure (as motivated in the following section), it is arguably easier to consider syntactic substitutions where possible. A simple yet useful case where correctness is not compromised is when complex formulas in the low-level theory appear in the same way everywhere, and so can be abstracted as a high-level atom. For example, suppose Δ_l mentions the atoms $\{p_1, \ldots, p_k, q, \ldots, r\}$. For simplicity, suppose Δ_l is in conjunctive normal form, that is, $\Delta_l = \phi_1 \wedge \ldots \wedge \phi_n$, where ϕ_i are clauses, and let λ be a clause only mentioning $\{q, \ldots, r\}$. Suppose for every i, either ϕ_i does not mention $\{q, \ldots, r\}$ or $\phi_i = \lambda \vee \psi_i$, where ψ_i is a clause only mentioning $\{p_1, \ldots, p_k\}$. In English: the symbols $\{q, \ldots, r\}$ do not appear in ϕ_i except as the clause λ. We can construct a high-level theory that replaces λ with a new atom t. So let Δ_h be exactly like Δ_l except that every instance of λ is replaced by t. Clearly the refinement mapping m maps t to λ and all other atoms to themselves. Further, let $\Pr(p_i, \Delta_h, w_h) = \Pr(p_i, \Delta_l, w_l)$ and $\Pr(t, \Delta_h, w_h) = \Pr(\lambda, \Delta_l, w_l)$. It is now not hard to see that (Δ_h, w_h) is not only a weak exact abstraction of (Δ_l, w_l) relative to m, but a weighted exact abstraction too.

Proposition 38. *Suppose $\Delta_h, w_h, \Delta_l, w_l, m$ are as above, and λ is satisfiable. Then (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l) relative to m.*

Proof. We will first prove that for every $M_h \in \mathit{Models}(\Delta_h)$ there is a $M_l \in \mathit{Models}(\Delta_l)$ such that $M_h \sim_m M_l$, and vice versa. So let M_h be any model of Δ_h. By assumption, λ is satisfiable, and suppose M' is one such satisfying assignment (note: M' is essentially a partial model for the atoms in $\mathit{Lang}(\Delta_l)$). Let M_l be exactly like M_h in interpreting $\{p_1, \ldots, p_k\}$, and for the atoms in $\{q, \ldots, r\}$, let M_l assign exactly as M' would. It now follows that for every atom $u \in \mathit{Lang}(\Delta_h)$, that is, $u \in \{p_1, \ldots, p_k, t\}$, we have that $M_h \models u$ iff $M_l \models m(u)$. The case of finding a high-level model for every low-level model is analogous. Thus, Δ_h is a sound and complete abstraction of Δ_l relative to m.

In terms of aligning probabilities, the case is immediate by construction. ∎

A slight variant of this idea could be used for abstracting multi-valued (or even continuous) random variables, provided the queries can be reasoned as Boolean variables (e.g., [48, 36, 5]). We demonstrate using an example below.

Example 39. Let us consider a random variable X that is uniformly drawn from $\{0, 1, \ldots, 9\}$ and suppose we are only interested in queries about $X \geq 8$ or its negation. In our terms, we could imagine $X \in \{0, 1, \ldots, 9\}$ to be represented using the formula $p_0 \vee \ldots \vee p_9$, and $X \geq 8$ to mean $p_8 \vee p_9$. More precisely, suppose the low-level theory (Δ_l, w_l) is one where Δ_l contains the following formulas:

- $p_0 \vee \ldots \vee p_9$
- $p_0 \equiv \neg(p_1 \vee \ldots \vee p_9), \ldots, p_9 \equiv \neg(p_0 \vee \ldots \vee p_8)$

and w_l is such that $\Pr(p_0, \Delta_l, w_l) = 0.1, \ldots, \Pr(p_9, \Delta_l, w_l) = 0.1$. Thus $\Pr(p_8 \vee p_9, \Delta_l, w_l) = 0.2$. A high-level theory Δ_h could be constructed as an abstraction containing the single atom q provided:

- m is a mapping $m(q) = p_0 \vee \ldots \vee p_7$;
- w_h is defined so that $\Pr(q, \Delta_h, w_h) = 0.8$ as a result of which $\Pr(\neg q, \Delta_h, w_h) = 0.2$.

It is now easy to see that (Δ_h, w_h) is a weighted exact abstraction of (Δ_l, w_l) relative to m.

What is interesting about this example in relation to Proposition 38 is that the $\lambda = p_0 \vee \ldots \vee p_7$ in question is abstracted as an atom q as usual, but we are using $\neg q$ to mean $p_8 \vee p_9$, which is not purely syntactical substitution. We are appealing to the property that $\Delta_l \models (p_0 \vee \ldots \vee p_7) \equiv \neg(p_8 \vee p_9)$ obtained by logical equivalence relative to Δ_l. Thus, like in Proposition 38, we are replacing a low-level formula λ with a high-level atom t but unlike that setting, we are replacing a low-level formula γ with $\neg t$ provided $\Delta_l \models \gamma \equiv m(\neg t)$. It is not hard to show that a correctness result also holds in this extended setting.

9.2. A Generic Search Algorithm

Let us now consider a generic search algorithm for deriving abstractions. We will begin with weak exact abstractions, as they are clearly simpler to treat than weighted exact abstractions. We return to the latter in a subsequent section.

The starting point here is that as input we are given the low-level theory (Δ_l, w_l). We are then interested in constructing a high-level theory and we assume to be also given the set of high-level predicates

$$P_h^1(x, \ldots, y), \ldots, P_h^k(x, \ldots, z)$$

from which Δ_h is to be constructed. If we now guess a (Δ_h, w_h) and a mapping m, we know from Definition 29 that testing for the following property would ascertain that the current guess is a weak exact abstraction:

$$\text{for all } \phi \in \mathit{Lang}(\Delta_h), \Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l) \qquad (\star)$$

Thus, a general schema might then look as follows:

Data: Low-level theory $(\Delta_l, w_l), \{P_h^1(x, \ldots, y), \ldots, P_h^k(x, \ldots, z)\}$
Result: success / failure
$(\Delta_h, w_h, m) = \{\}$
while *true* **do**
  Guess (Δ_h, w_h, m) that is different from previous guesses, where Δ_h only uses the mentioned predicates
  if $(\star)$ is true **then**
    return **success**
  if no more unique guesses **then**
    return **failure**
end

Algorithm 1: Guessing (Δ_h, w_h) and refinement mapping m

It is not hard to argue that the algorithm is correct:

Theorem 40. *Suppose Algorithm 1 returns* ***success****. Then* (Δ_h, w_h) *together with* m *is a weak exact abstraction. Suppose there are one or more weak exact abstractions only using* $\{P_h^1(x, \ldots, y), \ldots, P_h^k(x, \ldots, z)\}$*, then Algorithm 1 returns* ***success*** *with one such abstraction. Finally, Algorithm 1 returns* ***failure*** *iff there is no weak exact abstraction.*

Proof. Algorithm 1 returns **success** only when the current guess (Δ_h, w_h, m) satisfies $(\star)$. So soundness follows. Since we do exhaustive search, completeness follows. Analogously, Algorithm 1 returns **failure** iff none of the guesses from the exhaustive search satisfy $(\star)$. ∎

9.3. Effective Testability

In the above algorithm, the test $(\star)$ is challenging, because we would need to compute the probabilities for every formula. Even though we are assuming a finite vocabulary, we would still need to consider the set of all well-defined formulas over Boolean connectives of arbitrary but finite length, which is very large. Therefore, one might wonder if this test could be made easier, the most natural case being that of testing the probabilities of literals, which can be enumerated easily. Indeed, for a language with k predicates of arity w and a domain of size n, there will be at most $k \times n^w$ atoms, and so at most $2 \times k \times n^w$ literals. We will now show that such a result is possible, but we will need *separable* refinement mappings.

Definition 41. Suppose Δ_h and Δ_l are theories. We say a refinement mapping m from Δ_h to Δ_l is *separable* iff for every $\alpha \wedge \beta \in \mathit{Lang}(\Delta_h)$ such that α and β do not share (high-level) atoms, then it also the case $m(\alpha) \in \mathit{Lang}(\Delta_l)$ and $m(\beta) \in \mathit{Lang}(\Delta_l)$ do not share (low-level) atoms.

We do not expect this stipulation to be problematic in the least, and it seems entirely natural. For separable refinement mappings, we obtain the following property:

Theorem 42. *Suppose* (Δ_h, w_h) *and* (Δ_l, w_l) *are two theories and* m *is a separable refinement mapping from* Δ_h *to* Δ_l*. Suppose for all literals* $d \in \mathit{Lits}(\Delta_h)$*,* $\Pr(d, \Delta_h, w_h) = \Pr(m(d), \Delta_l, w_l)$*. Then for all* $\phi \in \mathit{Lang}(\Delta_h)$*,* $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$*.*

Proof. Proof by induction on the length of ϕ. The base case is immediate by definition, so assume for all $\phi \in \mathit{Lang}(\Delta_h)$ of length k, $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$. Consider $\phi \in \mathit{Lang}(\Delta_h)$ of length $k+1$, and let d be any atom mentioned in ϕ. Observe $\phi \equiv (\phi \wedge d) \vee (\phi \wedge \neg d)$. Let ϕ_i be ϕ but with every occurrence of d replaced by $i \in \{0, 1\}$ (denoting *false* and *true*) and every occurrence of $\neg d$ replaced by $1 - i$. Observe that $\phi \wedge d \equiv \phi_1 \wedge d$, that is, ϕ is simplified by setting all occurrences of d with 1 (*true*) and by setting all occurrences of $\neg d$ with 0 (*false*). Analogously, $\phi \wedge \neg d \equiv \phi_0 \wedge \neg d$. Because ϕ_i eliminates at least one literal from ϕ, its length is $\leq k$ and by hypothesis, $\Pr(\phi_i, \Delta_h, w_h) = \Pr(m(\phi_i), \Delta_l, w_l)$. Moreover, since d is not mentioned in ϕ_i, it follows that $\Pr(\phi_i \wedge d, \Delta_h, w_h) = \Pr(\phi_i, \Delta_h, w_h) \times \Pr(d, \Delta_h, w_h)$, and analogously for $\Pr(\phi_i \wedge \neg d, \Delta_h, w_h)$, $\Pr(m(\phi_i) \wedge m(d), \Delta_l, w_l)$, and so on.

From Theorem 17, we know that $\Pr(\alpha \vee \beta) = \Pr(\alpha) + \Pr(\beta) - \Pr(\alpha \wedge \beta)$. Let us now apply this to ϕ. So, $\Pr(\phi, \Delta_h, w_h)$

$= \Pr((\phi_1 \wedge d) \vee (\phi_0 \wedge \neg d), \Delta_h, w_h)$

$= \Pr(\phi_1 \wedge d, \Delta_h, w_h) + \Pr(\phi_0 \wedge \neg d, \Delta_h, w_h) - \Pr((\phi_1 \wedge d) \wedge (\phi_0 \wedge \neg d), \Delta_h, w_h)$

$= \Pr(\phi_1 \wedge d, \Delta_h, w_h) + \Pr(\phi_0 \wedge \neg d, \Delta_h, w_h)$ because $\Pr((\phi_1 \wedge d) \wedge (\phi_0 \wedge \neg d), \Delta_h, w_h) = 0$ owing to $(\phi_1 \wedge d) \wedge (\phi_0 \wedge \neg d)$ being inconsistent

$= \Pr(\phi_1, \Delta_h, w_h) \times \Pr(d, \Delta_h, w_h) + \Pr(\phi_0, \Delta_h, w_h) \times \Pr(\neg d, \Delta_h, w_h)$ because d or its negation is not mentioned in ϕ_i

$= \Pr(m(\phi_1), \Delta_l, w_l) \times \Pr(m(d), \Delta_l, w_l) + \Pr(m(\phi_0), \Delta_l, w_l) \times \Pr(m(\neg d), \Delta_l, w_l)$ by induction hypothesis

$= \Pr(m(\phi_1) \wedge m(d), \Delta_l, w_l) + \Pr(m(\phi_0) \times m(\neg d), \Delta_l, w_l)$ owing to m being separable (that is, $m(\phi_i)$ and $m(d)$ do not share atoms)

$= \Pr(m(\phi_1 \wedge d), \Delta_l, w_l) + \Pr(m(\phi_0 \wedge \neg d), \Delta_l, w_l) - \Pr(m(\phi_1 \wedge d) \wedge m(\phi_0 \wedge \neg d), \Delta_l, w_l)$ where $\Pr(m(\phi_1 \wedge d) \wedge m(\phi_0 \wedge \neg d), \Delta_l, w_l) = 0$ owing to its inconsistency (so is vacuously added)

$= \Pr(m(\phi), \Delta_l, w_l)$.

Therefore, $\Pr(\phi, \Delta_h, w_h) = \Pr(m(\phi), \Delta_l, w_l)$. ∎

Thus, when restricting to separable mappings, we can replace $(\star)$ in Algorithm 1 with:

$$\text{for all literals } d \in \mathit{Lits}(\Delta_h),\ \Pr(d, \Delta_h, w_h) = \Pr(m(d), \Delta_l, w_l) \qquad (\star\star)$$

In line 4, moreover, we would only be guessing separable mappings. It is then not hard to show that correctness still holds for the modified Algorithm 1, provided the existence of abstractions is stipulated as being limited to separable mappings.[10]

9.4. Effective Testability: Beyond Weak Abstractions

The constraints for weighted exact abstractions are clearly more involved, which, at first glance, would involve enumerating models and checking for isomorphic structures. Although this may be possible via techniques like knowledge compilation [17], one might wonder if simpler tests, like the ones asserted in $(\star\star)$ above, could also be obtained for weighted exact abstractions. What is really needed in addition to $(\star\star)$, of course, is a way to establish that Δ_h is a

[10] Observe that although our language is relational, most of the results essentially resort to ground theories. This is not uncommon in the literature on statistical relational learning (e.g., [25]), and as argued earlier, exploiting the relational structure for computational purposes [71] is orthogonal to the main thrust of this work. Nonetheless, if we were to make further assumptions about the theory, such as stipulating that all instances of a predicate occur with the same probability [3], we could perhaps simplify $(\star\star)$ further. It might suffice, for example, to simply check the probabilities of any arbitrary instance of the predicate to ascertain abstractions.

sound and complete abstraction of Δ_l relative to a mapping m. What we want to avoid, ideally, is a strategy for establishing the latter via tests involving the set of all well-defined formulas like in ($\star$). We now show that this is indeed possible.

Below, given a theory Δ, we sometimes think of it as a set of formulas $\{\phi_1, \ldots, \phi_k\}$, with the understanding that the theory is equivalent to $\phi_1 \wedge \ldots \wedge \phi_k$. We write $\phi_i \in \Delta$ to mean that ϕ_i is one of the formulas of that set Δ. We will also assume separable mappings for establishing our results:

Theorem 43. *Suppose Δ_h and Δ_l are logical theories, and m is a separable refinement mapping from Δ_h to Δ_l. Suppose for all $\phi \in \Delta_h$, $\Delta_l \models m(\phi)$. Then Δ_h is a sound abstraction of Δ_l relative to m.*

Proof. Suppose M_l is any model of Δ_l. Now, by assumption, for every $\phi \in \Delta_h$, $\Delta_l \models m(\phi)$, that is, $\Delta_l \models m(\Delta_h)$. Because $M_l \in \mathit{Models}(\Delta_l)$, $M_l \models m(\Delta_h)$.

Given all the atoms $p_1, \ldots, p_k$ of $\mathit{Lang}(\Delta_h)$, let $d_1, \ldots, d_k$ be the literals (say, $d_1 = p_1$, $d_2 = \neg p_2$, and so on) such that $M_l \models m(d_1) \wedge \ldots \wedge m(d_k)$. (By separability, note that for atoms $p, q \in \mathit{Lang}(\Delta_h)$, $m(p)$ and $m(q)$ do not share low-level atoms.) Let M_h be an interpretation of $\mathit{Lang}(\Delta_h)$ such that $M_h \models d_1 \wedge \ldots \wedge d_k$. By construction then $M_h \sim_m M_l$. By Theorem 6, for every $\phi \in \mathit{Lang}(\Delta_h)$, $M_h \models \phi$ iff $M_l \models m(\phi)$. Since $M_l \models m(\Delta_h)$, it follows that $M_h \models \Delta_h$. Since such a high-level model can be constructed for any $M_l \in \mathit{Models}(\Delta_l)$, Δ_h must be a sound abstraction of Δ_l relative to m. ∎

We now turn to completeness:

Theorem 44. *Suppose Δ_h and Δ_l are logical theories, and m is a separable refinement mapping from Δ_h to Δ_l. Suppose for all literals $d \in \mathit{Lang}(\Delta_h)$, if $d \wedge \Delta_h$ is satisfiable then so is $m(d) \wedge \Delta_l$. Then Δ_h is a complete abstraction of Δ_l relative to m.*

Proof. Suppose M_h is any model of Δ_h. Consider the formula $M_h^{\downarrow}$, which is the conjunction of literals that are true at M_h. In particular, suppose $M_h^{\downarrow} = d_1 \wedge \ldots \wedge d_k$. Clearly, $d_i \wedge \Delta_h$ is satisfiable, and so by assumption, $m(d_i) \wedge \Delta_l$ is satisfiable: let M_l^i be such a model where $m(d_i) \wedge \Delta_l$ is true. In particular, let $l_1^i \wedge \ldots \wedge l_{u_i}^i$ be a conjunction of literals true at M_l^i such that these literals mention all the atoms in $m(d_i)$ and only them. Put differently, $l_1^i \wedge \ldots \wedge l_{u_i}^i \models m(d_i)$, and we can also see $l_1^i \wedge \ldots \wedge l_{u_i}^i$ as a partial interpretation for Δ_l. When we consider such partial interpretations for $i \neq j$, by separability it follows that $m(d_i)$ and $m(d_j)$ do not share atoms, so $L^i = l_1^i \wedge \ldots \wedge l_{u_i}^i$ and $L^j = l_1^j \wedge \ldots \wedge l_{u_j}^j$ do not share atoms, and thus are consistent with each other. In other words, we now have the partial interpretation $L^1 \wedge \ldots \wedge L^k$ of Δ_l such that for each i: (a) $L^i \models m(d_i)$, (b) L^i mentions all and only the atoms in $m(d_i)$. Let M_l be an interpretation of Δ_l where $L^1 \wedge \ldots \wedge L^k$ holds. Then for every atom $p \in \mathit{Lang}(\Delta_h)$, $M_h \models p$ iff $M_l \models m(p)$, so $M_h \sim_m M_l$. ∎

Putting it all together, in Algorithm 1, to test whether a guess (Δ_h, w_h, m) is a weighted exact abstraction, we would need three checks:

1. for all $\phi \in \Delta_h, \Delta_l \models m(\phi)$;
2. for all literals $d \in \mathit{Lang}(\Delta_h)$, if $d \wedge \Delta_h$ is satisfiable then so is $m(d) \wedge \Delta_l$; and
3. ($\star\star$).

So of course the test involves more computations than for weighted exact abstractions, but as we discuss above, we additionally obtain logical alignment should that be desired.

9.5. Constraining search

Algorithm 1 is discussed at a general level, which is deliberate, but that also means that no commitment has been made yet on how to define and constrain the search space.[11] The case of formula replacements was discussed in Section 9.1. Below, we discuss two other strategies. We reiterate that to make the process of deriving abstractions effective some combination of such strategies along with a suitably restricted fragment is likely needed.

[11] One could imagine that if the high-level predicates are not provided as input for Algorithm 1, we might then parameterize the algorithm by providing a vocabulary bound k and an arity bound z, and attempt to guess a high-level theory from k predicates of maximum arity z. If that fails, the bound could be incremented.

Partial knowledge: The user might be in a position to suggest (and constrain) the space of possible mappings and the syntactical form of the high-level theory. For example, for some unknown Δ_h over the predicates $\{P_h^1(x,\ldots,y),\ldots,P_h^k(x,\ldots,z)\}$, suppose the user decides to use the same domain as Δ_l. Suppose she provides partial information about the refinement mapping: let m' be a mapping from atoms $p \in L$ to θ_p, where $\theta_p \in \mathit{Lang}(\Delta_l)$ and $L \subseteq \mathit{Lits}(\Delta_h)$. Let us further assume the user provides partial knowledge of the sentences in Δ_h, say Δ'_h. Then line 4 of Algorithm 1 would guess functions m that extends m' (that is, possible completions of m), and line 3 would guess Δ_h such that $\Delta'_h \subseteq \Delta_h$. It would then follow that with exhaustive search, correctness would still be shown to hold, provided the existence of abstractions is stipulated as being limited to extensions of the partial knowledge (that is, the partial knowledge is assumed to be correct in the simplest case). If the user were to provide examples $\{e_1,\ldots,e_k\}$ instead of a sub-theory Δ'_h, we might take an approach akin to inductive logic programming [53]. That is, we first define a syntactic bias, that is, a hypothesis space $\mathcal{H}$, and find a $\Delta_h \in \mathcal{H}$ satisfying a semantic bias (say, $\Delta_h \models e_1 \wedge \ldots \wedge e_k$.) In this case, if Algorithm 1 were to return **success**, soundness is immediate, but further investigations are needed to show that an appropriate $\mathcal{H}$ also promises completeness.

Decomposability: Rather than searching for abstractions by treating Δ_l as a monolithic entity, a pragmatic alternative is possible when the low-level theory is *decomposable*: that is, it is a set of sentences that are logically independent of each other. Then, we can identify *local* abstractions and compose those to obtain a *global* solution. Consider the case where $\Delta_l = \phi_1 \wedge \ldots \wedge \phi_k$, where ϕ_i does not share atoms with ϕ_j for all $i \neq j$. Such decompositions may appear naturally when a joint distribution is characterized over a set of disjoint Markov networks [19, 58, 5], or may be obtained by knowledge compilation [17] from a more involved theory. Recent advances in tractable learning [28], which have their roots in knowledge compilation, also identify clusters of random variables that are independent of each other. It follows that $\Pr(\phi_i \wedge \phi_j) = \Pr(\phi_i) \times \Pr(\phi_j)$ and $\Pr(\phi_i \vee \phi_j) = \Pr(\phi_i) + \Pr(\phi_j) - \Pr(\phi_i) \times \Pr(\phi_j)$. Basically then we can identify $\Delta_h = \psi_1 \wedge \ldots \wedge \psi_k$, where ψ_i abstracts ϕ_i, and shares the structural restriction that ψ_i and ψ_j do not share atoms for all $i \neq j$. The abstraction search would be limited locally to ϕ_i, that is, for example, wrt each low-level Markov network.

Fortunately, we are able to show that this intuitive idea is correctness preserving. We prove the case of complete abstractions, and the other cases are analogous.

Theorem 45. *Suppose ψ_i is a complete abstraction of ϕ_i relative to m_i. Suppose ϕ_i and ϕ_j do not share atoms for all $i \neq j$, and analogously, ψ_i and ψ_j do not share atoms for all $i \neq j$. Then $\Delta_h = \psi_1 \wedge \ldots \wedge \psi_k$ is a complete abstraction of $\Delta_l = \phi_1 \wedge \ldots \wedge \phi_k$ relative to $m = m_1 \wedge \ldots \wedge m_k$ (that is, the composite mapping obtained by extending m_1 to include the vocabulary and mapping of m_2, which is then extended for m_3, and so on.)*

Proof. Suppose M_h is any model of Δ_h. Consider the formula $M_h^{\downarrow}$, which is the conjunction of literals that are true at M_h. Consider that $M_h^{\downarrow}$ can be written as $h_1^{\downarrow} \wedge \ldots \wedge h_k^{\downarrow}$, where:

- h_i is an interpretation for $\mathit{Lang}(\psi_i)$;
- following our notation, $h_i^{\downarrow}$ is a conjunction of literals; and so
- $h_i^{\downarrow}$ only mentions the atoms from $\mathit{Lang}(\psi_i)$.

By construction, since M_h is a model of Δ_h, h_i must be a model of ψ_i. By assumption, there is a model l_i of ϕ_i such that h_i is isomorphic to l_i relative to m_i. Then let M_l be the model corresponding to the formula $M_l^{\downarrow} = l_1^{\downarrow} \wedge \ldots \wedge l_k^{\downarrow}$. By construction, M_l must be a model of Δ_l, and moreover, from the isomorphism that holds for h_i and l_i relative to m_i, $M_h \sim_m M_l$. ∎

9.6. Complexity

For a propositional formula, finding a satisfying assignment (SAT) is the prototypical NP-complete problem, whereas enumerating the set of satisfying assignments, i.e., model counting or #SAT, is the prototypical #P-complete problem [1, 59, 70]. Weighted model counting, which extends #SAT in according weights to formulas, has a polynomial reduction to the unweighted formulation [1].

So let us consider a worst-case analysis here. To establish that (Δ_h, w_h) is a sound and complete abstraction of (Δ_l, w_l) relative to m we may appeal to Definition 9 and Definition 12. This means that we would need to enumerate

the high-level models, which #P-hard in the worst case, and enumerate the low-level models, which is also #P-hard in the worst case. Given two models M_h and M_l, establishing $M_h \sim_m M_l$ requires us to enumerate the atoms in the high-level language, which is linear in the vocabulary of the high-level language, to test whether the atom is satisfied (a look-up operation). Since model counting is computationally challenging, the worst-case analysis for establishing abstractions is perhaps not surprising.

If the problem were approached not by enumerating models per se but rather by computing the probabilities of formulas, the situation is, of course, still the same. Recall that given n propositions, we can have exponentially many formulas in general, as discussed before. Thus, the most effective way to establish abstractions is by leveraging the ideas on testability from above, and simply look to testing probabilities for the literals in the high-level language. Note that the number of literals is linear in the vocabulary of the language, and so for each literal, we would compute the probability of that literal wrt the weighted theory. Computing that probability is #P-hard in the worst case, but the number of calls to the #SAT oracle would be linear (in the vocabulary).

It would be interesting to investigate whether all of this can be made more effective by appealing to strategies such as knowledge compilation [17], approximate model counting [11], caching of subcomputations [1] and/or tractable models enforcing certain factorizations [40, 44, 13].

10. Related Work and Discussion

Abstraction is a major topic in knowledge representation [30, 23, 61, 2]. The idea of establishing mappings between models to yield a semantic theory for abstraction owes its origin to works such as that of Milner [50]. But formal treatments have been mostly restricted to categorical and non-probabilistic domains. Nonetheless, there have been various developments in different communities, and we discuss the lineage in more detail below.

Knowledge representation and automated planning: Our framework here is inspired by, and indeed builds on the proposal by Banihashemi et al. [2], where isomorphism as well as sound and complete abstractions are investigated for (non-probabilistic) situation calculus agent programs. Here, we sought to extend those ideas to establish probabilistically interesting properties for probabilistic models, including probabilistic relational models (PRMs). For this, we motivated the notion of unweighted abstractions, which roughly corresponds (at the level of satisfaction and entailment) to the categorical setting [2]. But by piggybacking on these properties, weighted abstractions and evidence incorporation were motivated and formulated. We then identified the relationship of that framework to a purely stochastic one (weak exact abstractions), and further studied the automatic derivation of abstractions. Our observations about decomposability, among other things, need not be limited to stochastic models and may also be applicable in the categorical setting.

We refer interested readers to the discussion by Banihashemi et al. [2] for a comprehensive account on the use of abstraction in knowledge representation and automated planning (e.g., hierarchical planning). A particularly interesting direction in this landscape is the work of Giunchiglia and Walsh [30], where operations on abstractions are studied at the level of the logical theories. We suspect these results can be generalized (and adapted) to our framework with some effort.

Program synthesis and verification: In the area of program verification, static analysis and abstraction interpretations are commonplace to test the correctness of programs and probabilistic programs. See McIver et al. [47] for a book length treatment, for example. A number of additional concerns present themselves in a programmatic setting, including the branching in the presence of stochastic primitives, and stochastic transitions between program states. The motivation then is to deduce sound abstractions for verifying correctness (e.g., termination) properties. While some of these works do not consider abstractions themselves to be probabilistic, the developments are related to our goals. Thus, it would be interesting to study how ideas and techniques from the program analysis literature can be carry over to our framework, and vice versa. For example, Zhang et al. [73] consider statistical properties of program behavior to advise abstractions, Sharma et al. [65] relate verification to the learnability of concepts, Holtzen et al. [36] study abstract predicates for loop-free probabilistic programs, and Monniaux [51] defines abstract representations for probabilistic program path analysis. On the semantical front, Cousot and Monerau [15] present a detailed and careful analysis for reasoning about (probabilistic) nondeterminism in programs, but as argued by Holtzen et al. [36], they do consider the abstractions themselves to be probabilistic structures. This is precisely the focus of Holtzen et al. [36], where they want to abstract loop-free probabilistic programs as possibly simpler and smaller probabilistic programs.

These latter programs may then be amenable to automated verification, for example. Roughly, the idea is to transform a concrete program (in our terms: low-level) to an abstract program (in our terms: high-level). So this work is closest in spirit to our thrust. They mainly motivate a notion of sound probabilistic over-approximation, with the intent of capturing a distribution over feasible states. In a sense, this is akin to Definition 25, but without any stipulation of logical alignment, so it is *weak*. In follow up work, Holtzen et al. [37] introduce the notion of distributional soundness, which asserts that the probability of a high-level event is equal to the probability of the low-level event. Thus, in a sense, this is akin to weak exact abstractions. Finally, they also investigate a strategy for finding high-level programs that roughly corresponds to replacing atomic formulas in the low-level program with an appropriate high-level random variable [37, Algorithm 1]. That procedure is similar in spirit to our generic search procedure, in the sense of requiring exponential search in the worst case, but considerably easier given the predefined structural syntax of the sought after abstraction. They discuss an implementation that also leverages many state-of-the-art techniques from the program verification community, such as counterexample-guided refinement [14]. Put differently, the setting of loop-free probabilistic programs built from Bernoulli or other univariate distributions is often much more constrained than a first-order language in that the grammar only allows conjunctions of positive statements (as in most loop-free sequential programs), and programs allow the use of the predicate transformer semantics [20] to abstract atomic assertions. Our setting is more general, in that: (a) high-level abstractions may map onto arbitrarily complex well-defined low-level formulas, (b) along with weak abstractions, we also motivate theory alignment (such as Definition 27), and (c) the treatment of evidence at the first-order/logical level allows for a richer perspective. Nonetheless, restricted settings such as the one investigated for probabilistic programs may represent cases that offer reasonable expressiveness/tractability tradeoffs. In that regard, refining schemas such as Algorithm 1 to also leverage state-of-the-art techniques from the program verification community may identify other interesting fragments.

Finally, in the context of inductive logic programming and meta-interpretative learning, there is a long history of predicate invention and learning abstract programs [16] and their affect on human comprehensibility [62]. Although the semantic constraints for abstracting is somewhat different, it is possible that our setup regarding partial knowledge in Section 9 could be realized using such methods.

Probabilistic logical modeling: In the PRM literature, a few recent developments seem to be close in spirit to abstraction. In the work of Sen et al. [63], the question of making inference more efficient in classes of probabilistic databases (PDBs) that share certain structural properties is investigated. Roughly, what they are after is a possibly "compressed" PDB that answers queries exactly as would the original PDB in the manner that inference computations are not repeated for the shared features. Of course, our framework differs in that the high-level and low-level theory do not need to have any structural similarities. Moreover, if they do share structural similarities, at this point, we disregard the issue of how probabilistic computations can be made efficient, as this is somewhat orthogonal to the main thrust of the paper. We suspect, under some conditions, it might be possible to show that classes of PDBs with shared features may correspond to abstractions, but conversely, reiterating the point above, simply because Δ_h and Δ_l are abstractions need not imply that they share structural features. Along these lines, a recent thrust in PRMs is the question of how to make inference more efficient by exploiting the relational vocabulary, referred to as "lifted reasoning" [71, 64]. This is justifiably sometimes referred to as a type of "abstraction" [46]. There seem to be two implications for our work. The first is computational: verifying that Δ_h is an abstraction of Δ_l could be made more efficient by appealing to lifted reasoning. (This is, as argued elsewhere, somewhat orthogonal to the main thrust of the paper.) The second revisits our observations about compressed PDBs [63]. One could, for example, see a non-ground PRM as the high-level abstraction of the low-level ground PRM, in that none of the domain constants are explicitly mentioned in the former. So, in that sense, a non-ground PRM would turn out to be an abstraction of a ground PRM, but simply because Δ_h and Δ_l are abstractions need not imply that they share the same vocabulary. For the future, we hope to study the connections between these strands of work and abstraction in more detail, so as to attempt to formalize these intuitions.

On the topic of reasoning, our framework has some overlap with the principles of *representation-independent* probabilistic inference [34]. Here, one is usually interested in the computed conditional queries not being different if the knowledge base is represented differently, (say) using an abstract logical language. So, Halpern and Koller [34] motivate a notion of correctness where if a query ϕ follows a knowledge base Δ, it should also be the case that $f(\phi)$ follows from $f(\Delta)$ where f is a mapping from one representation to another. Then, a *robust* inference procedure is one that respects semantically justifiable properties such as those in Theorem 17 for reasonable mappings. While there is

some similarity at first glance, there is a crucial conceptual difference: as already argued by Halpern and Koller [34], unlike the work in representation-independent inference, the two representations are not required to be equivalent in an exercise on abstraction, because, by definition, an abstraction ignores irrelevant facts. The technical thrust is also different in our work, such as the identification of weighted exact vs weak exact abstractions, the handling of evidence, and our analysis on generating abstractions. Following the work of Halpern and Koller [34], a broader treatment is given by Jaeger [38], where the notion of representation independence is studied for non-classical consequence more generally, of which probability measures is a special case. Very similar in spirit to our own work as well as the categorical setting that we build on [2], the purely logical question of when two sentences represent the same information is considered first, prompting a definition that is virtually identical to unweighted sound and complete abstractions. As argued above for the case of Halpern and Koller [34], there are numerous differences in terms of technical thrust, however. In our work, for example: (a) abstractions were analyzed at the level of soundness and completeness; (b) weighted abstractions were derived by piggybacking on constraints noted in the unweighted setting; (c) we investigated the difference between probabilistic alignment in the presence and absence of logical alignment; (d) we considered the incorporation of evidence; and (e) we identified properties for the verification and generation of abstraction.

We note that although much of our discussion has focused on formal accounts addressing notions of abstraction, there has been considerable effort on the pragmatic side to connect high-level (mostly logical) concepts with low-level (mostly probabilistic) variables. Research on symbol acquisition and symbol grounding and its relation to high-level planning [45], and symbolic concept learning from sub-symbolic methods [9] are prominent examples of such work. More generally, many commonsensical knowledge base and rule extraction techniques often define concepts at different levels of granularity, and thus are also relevant from a pragmatic viewpoint [7, 22, 57].

Statistics: When establishing the alignment between the high-level and low-level theory, we looked solely at the marginal probabilities between two discrete probability distributions. Summarizing distributions is a standard problem in statistics, and the use of means and moments is common [54]; it is an interesting question whether such constructs could be useful for defining and/or deriving abstractions (in our sense).

A more standard case of statistical abstraction is when continuous distributions are cast as discrete events by appealing to the cumulative distribution function. We touched upon this in Section 9.1, and we note that such ideas have been used for inference in continuous domains via a generalized variant of weighted model counting [5], and for probabilistic program abstractions [36].

In causal modeling, mapping macro and micro level events is a long-standing concern, which correspond in our terms to high-level and low-level models. In recent work, for example, Rubenstein et al. [60] study consistency between micro and macro-level random variables via structural equation models, and so are close in spirit to abstractions.

Let us conclude this section with some observations. Firstly, despite the tremendous amount of attention that abstraction has received, the framework presented here is useful for a number of reasons: (a) it is downward compatible with the categorical (and first-order) setting [2]; (b) it identifies probabilistic alignment together with a notion of logical alignment, the former closely mirroring the analysis on probabilistic program abstractions [36, 37]; and (c) it seems to agree with the intuitions regarding representation-independence probabilistic inference [34]; and (d) it can leverage the advances in weighted model counting, including lifted reasoning. Let us now reflect on a few critical points.

It is interesting to note that although the level of generality of our framework allows us to easily draw comparisons to results from the categorical literature and representation independence, which is useful from a theoretical standpoint, it may mean that the formal results are somewhat removed from the concerns of high-level modeling languages. For example, a major issue with PRMs is ensuring that the ground model is acyclic so that inference and sampling methods can be designed easily. To understand how those issues carry over to abstraction algorithms, we need to understand how high-level PRMs can be designed that ensure that such properties are not lost on abstraction if present in the low-level PRM. Conversely, even if the low-level PRM does not enjoy effective inference properties, can a high-level PRM be obtained that is amenable to those properties? In this work, since our focus was on understanding the semantical properties of abstractions, such concerns are orthogonal but they may impact our choice for guessing an appropriate high-level theory.

To that end, existing empirical observations on abstraction-type frameworks are somewhat mixed. Early work on categorical abstractions [30], for example, noted that " ... shows that there are situations where abstraction saves

time but also situations where it results in less efficiency." In a similar vein, as discussed by Holtzen et al. [37], factor graph abstractions for probabilistic programs do not always maintain structural decompositions. On the one hand, if one takes a worst-case view of the inference problem, it immediately follows that given reasoning with (say) n atoms in Δ_l versus $m \ll n$ atoms in Δ_h, the latter seems preferable. For the case of loop-free sequential probabilistic programs built from Bernoulli random variables and conditional statements, Holtzen et al. [37] also empirically show that abstractions can be effective. Moreover, as already mentioned, abstraction is a very successful strategy in the verification communities. Thus, we think there is a broader question of choosing the most reasonable abstraction, one that is indeed amenable to effective inference, perhaps more so than the low-level theory.

A parallel concern is about the choosing of a vocabulary that offers maximum comprehensibility [62]. A minor observation to be made here is that if comprehensibility is all we care about (as opposed to being concerned about both comprehensibility and tractability), then it is not clear that one would need to fully abstract a theory. Indeed, we could introduce/invent definitional or semi-definitional predicates of the form $P(\vec{x}) \supset \phi(\vec{x})$ or $\phi(\vec{x}) \supset P(\vec{x})$, where $\phi(\vec{x})$ corresponds to low-level information, the idea being that the user is only exposed to instances of P. Such a construction would invariably increase the size of the theory, but we would be taking steps toward comprehensibility by designing predicates to correspond closely to the user's vocabulary. In fact, the user could be exposed to a combination of high-level and low-level predicates to provide explanations of a suitable granularity.

Such concerns, of course, affect all frameworks on abstraction, and are not unique to our endeavor. A semantic theory of correctness such as the one considered in this paper, in that regard, would indeed be at the level of interpretations and not necessarily in the language of the user-specific domain knowledge, perhaps analogous to a theory on program correctness. We would not expect a mathematical analysis on correct programs to necessarily involve the syntactic constructs of the programming language represented as is, but rather in terms of suitable mathematical objects that captures its computational underpinnings (e.g., variable assignment, recursion). In this sense, a semantical theory on abstraction is (in the non-technical sense) *abstract*. Owing to our observations and building on this theory, we do hope that it becomes possible to define and inspect the choosing of effective abstractions in the future.

11. Conclusions

In this work, we were motivated in the development of a formal framework for abstractions, based on isomorphisms between models, where atoms in a high-level theory can be mapped to complex formulas at the low-level. From that, we developed a number of accounts of abstraction, as well as the handling of low-level evidence, all of which motivated some observations about how abstractions can be derived automatically.

Given the increasing interest in abstraction for statistical AI, we hope our framework will be helpful in developing probabilistic abstractions for increased clarity, modularity and tractability, and perhaps interpretability.

[1] F. Bacchus, S. Dalmao, and T. Pitassi. Solving #SAT and Bayesian inference with backtracking search. *J. Artif. Intell. Res. (JAIR)*, 34:391–442, 2009.

[2] B. Banihashemi, G. De Giacomo, and Y. Lespérance. Abstraction in situation calculus action theories. In *AAAI*, pages 1048–1055, 2017.

[3] P. Beame, G. Van den Broeck, E. Gribkoff, and D. Suciu. Symmetric weighted first-order model counting. In *PODS*, pages 313–328. ACM, 2015.

[4] V. Belle. Open-universe weighted model counting. In *AAAI*, pages 3701–3708, 2017.

[5] V. Belle, A. Passerini, and G. Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *IJCAI*, pages 2770–2776, 2015.

[6] V. Belle, G. Lakemeyer, and H. J. Levesque. A first-order logic of probability and only knowing in unbounded domains. In *AAAI*, pages 893–899, 2016.

[7] F. Bianchi, M. Palmonari, P. Hitzler, and L. Serafini. Complementing logical reasoning with sub-symbolic commonsense. In *International Joint Conference on Rules and Reasoning*, pages 161–170. Springer, 2019.

[8] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *UAI*, pages 115–123, 1996.

[9] E. Cambria, S. Poria, D. Hazarika, and K. Kwok. Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*, pages 1795–1802, 2018.

[10] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, pages 1306–1313, 2010.

[11] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi. Distribution-aware sampling and weighted model counting for SAT. In *AAAI*, pages 1722–1730, 2014.

[12] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[13] A. Choi, D. Kisa, and A. Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 121–132. Springer, 2013.

[14] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In *International Conference on Computer Aided Verification*, pages 154–169. Springer, 2000.

[15] P. Cousot and M. Monerau. Probabilistic abstract interpretation. In *European Symposium on Programming*, pages 169–193. Springer, 2012.

[16] A. Cropper and S. H. Muggleton. Learning higher-order logic programs through abstraction and invention. In *IJCAI*, pages 1418–1424, 2016.

[17] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17: 229–264, 2002.

[18] G. Dedre and H. Christian. Analogy and abstraction. *Topics in Cognitive Science*, 9(3):672–693, 2017.

[19] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):380–393, 1997.

[20] E. W. Dijkstra and C. S. Scholten. *Predicate calculus and program semantics*. Springer Science & Business Media, 2012.

[21] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610, 2014.

[22] M. Dragoni, S. Poria, and E. Cambria. Ontosenticnet: A commonsense ontology for sentiment analysis. *IEEE Intelligent Systems*, 33(3):77–85, 2018.

[23] K. Erol, J. Hendler, and D. S. Nau. Complexity results for htn planning. *Annals of Mathematics and Artificial Intelligence*, 18(1):69–93, 1996.

[24] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994. ISSN 0004-5411.

[25] D. Fierens, G. Van den Broeck, I. Thon, B. Gutmann, and L. De Raedt. Inference in probabilistic logic programs using weighted CNF's. In *UAI*, pages 211–220, 2011.

[26] H. Gaifman. Concerning measures in first order calculi. *Israel Journal of Mathematics*, 2(1):1–18, 1964.

[27] A. Garfinkel. *Forms of explanation: Rethinking the questions in social theory*. Yale University Press New Haven, 1981.

[28] R. Gens and D. Pedro. Learning the structure of sum-product networks. In *ICML*, pages 873–880, 2013.

[29] L. Getoor and B. Taskar, editors. *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[30] F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial intelligence*, 57(2-3):323–389, 1992.

[31] C. P. Gomes, A. Sabharwal, and B. Selman. Model counting. In *Handbook of Satisfiability*. IOS Press, 2009.

[32] D. Gunning. Explainable artificial intelligence (xai). Technical report, DARPA/I20, 2016.

[33] J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003. ISBN 0262083205.

[34] J. Y. Halpern and D. Koller. Representation dependence in probabilistic inference. *Journal of Artificial Intelligence Research*, 21:319–356, 2004.

[35] D. Heckerman, C. Meek, and D. Koller. Probabilistic models for relational data. Technical report, Technical Report MSR-TR-2004-30, Microsoft Research, 2004.

[36] S. Holtzen, T. Millstein, and G. Van den Broeck. Probabilistic program abstractions. In *UAI*, 2017.

[37] S. Holtzen, G. Broeck, and T. Millstein. Sound abstraction and decomposition of probabilistic programs. In *ICML*, pages 2004–2013, 2018.

[38] M. Jaeger. Representation independence of nonmonotonic inference relations. In *KR*, pages 461–472. Morgan Kaufmann Publishers Inc., 1996.

[39] M. Jaeger. Reasoning about infinite random structures with relational bayesian networks. In *KR*, pages 570–581, 1998.

[40] A. Jha, V. Gogate, A. Meliou, and D. Suciu. Lifted inference seen from the other side: The tractable features. In *NIPS*, pages 973–981, 2010.

[41] G. Jorland. Idealization and transformation. *Idealization VI: Idealization in economics*, pages 265–275, 1994.

[42] K. Kersting, S. Natarajan, and D. Poole. Statistical relational AI: Logic, probability and computation. 2011.

[43] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[44] D. Koller, A. Levy, and A. Pfeffer. P-classic: a tractable probablistic description logic. In *Proc. AAAI / IAAI*, pages 390–397, 1997.

[45] G. Konidaris, L. Kaelbling, and T. Lozano-Perez. Symbol acquisition for probabilistic high-level planning. In *IJCAI*, pages 3619–3627, 2015.

[46] S. Lüdtke, M. Schröder, F. Krüger, S. Bader, and T. Kirste. State-space abstractions for probabilistic inference: A systematic review. *Journal of Artificial Intelligence Research*, 63:789–848, 2018.

[47] A. McIver, C. Morgan, and C. C. Morgan. *Abstraction, refinement and proof for probabilistic systems*. Springer Science & Business Media, 2005.

[48] S. Michels, A. Hommersom, and P. J. Lucas. Approximate probabilistic inference with bounded error for hybrid probabilistic logic programming. In *IJCAI*, pages 3616–3622. AAAI Press, 2016.

[49] B. Milch, B. Marthi, D. Sontag, S. J. Russell, D. L. Ong, and A. Kolobov. Approximate inference for infinite contingent bayesian networks. In *AISTATS*, pages 238–245, 2005.

[50] R. Milner. *Communication and Concurrency. International series in computer science*. Prentice hall Englewood Cliffs, 1989.

[51] D. Monniaux. An abstract monte-carlo method for the analysis of probabilistic programs. In *ACM SIGPLAN Notices*, volume 36, pages 93–101. ACM, 2001.

[52] D. Montague and B. Rajaratnam. Graphical Markov models for infinitely many variables. *ArXiv e-prints*, Jan. 2015.

[53] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

[54] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[55] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28, 2012.

[56] S. Penkov and S. Ramamoorthy. Explaining transition systems through program induction. *CoRR*, abs/1705.08320, 2017.

[57] G. Pilato, A. Augello, G. Vassallo, and S. Gaglio. Sub-symbolic semantic layer in cyc for intuitive chat-bots. In *International Conference on Semantic Computing (ICSC 2007)*, pages 121–128. IEEE, 2007.

[58] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.

[59] D. Roth. On the hardness of approximate reasoning. *Artif. Intell.*, 82(1-2):273–302, 1996.

[60] P. K. Rubenstein, S. Weichwald, S. Bongers, J. M. Mooij, D. Janzing, M. Grosse-Wentrup, and B. Schölkopf. Causal consistency of structural equation models. *arXiv preprint arXiv:1707.00819*, 2017.

[61] L. Saitta and J.-D. Zucker. *Abstraction in artificial intelligence and complex systems*, volume 456. Springer, 2013.

[62] U. Schmid, C. Zeller, T. Besold, A. Tamaddoni-Nezhad, and S. Muggleton. How does predicate invention affect human comprehensibility? In *International Conference on Inductive Logic Programming*, pages 52–67. Springer, 2016.

[63] P. Sen, A. Deshpande, and L. Getoor. Exploiting shared correlations in probabilistic databases. *Proceedings of the VLDB Endowment*, 1(1):809–820, 2008.

[64] P. Sen, A. Deshpande, and L. Getoor. Bisimulation-based approximate lifted inference. In *UAI*, pages 496–505. AUAI Press, 2009.

[65] R. Sharma, S. Gupta, B. Hariharan, A. Aiken, and A. V. Nori. Verification as learning geometric concepts. In *International Static Analysis Symposium*, pages 388–411. Springer, 2013.

[66] P. Singla and P. M. Domingos. Markov logic in infinite domains. In *UAI*, pages 368–375, 2007.

[67] S. Sreedharan, S. Srivastava, and S. Kambhampati. Hierarchical expertise level modeling for user specific contrastive explanations. In *IJCAI*, pages 4829–4836, 2018.

[68] D. Suciu, D. Olteanu, C. Ré, and C. Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3 (2):1–180, 2011.

[69] W. F. Trench. *Introduction to real analysis*. Prentice Hall, 2003.

[70] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3): 410–421, 1979.

[71] G. Van den Broeck. *Lifted Inference and Learning in Statistical Relational Models*. PhD thesis, KU Leuven, 2013.

[72] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proc. Int. Conf. on Management of Data*, pages 481–492, 2012.

[73] X. Zhang, X. Si, and M. Naik. Combining the logical and the probabilistic in program analysis. In *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, MAPL 2017, pages 27–34, New York, NY, USA, 2017. ACM.