Excursions in first-order logic and probability: infinitely many random variables, continuous distributions, recursive programs and beyond[⋆]

Vaishak Belle[0000−0001−5573−8465]

University of Edinburgh, UK
Alan Turing Institute, UK
vbelle@ed.ac.uk

Abstract. The unification of the first-order logic and probability has been seen as a long-standing concern in philosophy, AI and mathematics. In this talk, I will briefly review our recent results on revisiting that unification. Although there are plenty of approaches in communities such as statistical relational learning, automated planning, and neuro-symbolic AI that leverage and develop languages with logical and probabilistic aspects, they almost always restrict the representation as well as the semantic framework in various ways that does not fully explain how to combine first-order logic and probability theory in a general way. In many cases, this restriction is justified because it may be necessary to focus on practicality and efficiency. However, the search for a restriction-free mathematical theory remains ongoing. In this article, we discuss our recent results regarding the development of languages that support arbitrary quantification, possibly infinitely many random variables, both discrete and continuous distributions, as well as programming languages built on top of such features to include recursion and branching control.

Keywords: First-order logic · Probabilistic logic · Programs

1 Introduction

The unification of the logic and probability has been seen as a long-standing concern in philosophy and mathematical logic, going back to Carnap [14] and Gaifman [23], at least in terms of early rigorous algebraic studies. In artificial intelligence, starting from Nilsson [47], Bacchus [2] and Halpern [25], a wide range of formalisms encompassing various first-order logical features have been proposed. In so much as a probabilistic underpinning provides the gateway for incorporating probabilistic induction, areas such as statistical relational learning [50], inductive logic programming [45] and neuro-symbolic AI [27], are promising candidates for unifying deduction, noisy sensory observations and association-based pattern learning.

[⋆] This material introduces the topic of my keynote at the *18th Edition of the European Conference on Logics in Artificial Intelligence* (**JELIA**), September 20-22, 2023. This research was partly supported by a Royal Society University Research Fellowship, UK, and partly supported by a grant from the UKRI Strategic Priorities Fund, UK to the UKRI Research Node on Trustworthy Autonomous Systems Governance and Regulation (EP/V026607/1, 2020–2024).

1.1 What's missing from a first-order viewpoint?

From a knowledge representation viewpoint, however, especially in the context of reasoning about first-order knowledge – which by design must resolve issues around quantification, and an arbitrary (possibly infinite) domain of discourse – there is very little work beyond the initial study by Halpern [25]. This is despite the fact that first-order logic is widely acknowledged to be very important for capturing common sense and reasoning about generalized assertions [38, 37] in a way that humans intuitively seem to be able to do so.

There are many practical dimensions to this concern. For example, even though data that automated systems would encounter is finite, we still want knowledge such as "the father of a father is a grandfather" to be applicable to all humans in the universe. A robot might encounter an object and it is reasonable to assume that there are objects that are potentially occluded by the first object, and behind it. So the mental model of the robot should allow for the possibility that an unknown number of objects are behind the first object.

This issue, of course, has not gone unnoticed by the community. There are plenty of works that explore concepts such as open-world modeling – in contrast to the *closed-world assumption* that assumes the set of objects is fixed and finite in advance [4, 42]. But most solutions in the literature make a number of assumptions in terms of the modeling and the reasoning capabilities of the frameworks. This is justified from the point of view of practicality, but it leaves open what *a general mathematical theory for first-order logic and probability* looks like. Such a theory should allow, in the least:

1. infinite domains and arbitrary quantification from a first-order viewpoint;
2. discrete, continuous and mixed discrete-continuous distributions from a probability theory viewpoint.

The logical framework, understandably, should provide a calculus where we can express sentences that mix logical and probabilistic assertions. Moreover, these may be provided at any level of specificity by the modeler [3, 9] such that the model theory allows one to reason about entailments in a way that adheres to the axioms of first-order logic as well as probability theory.

1.2 The story does not get easier with actions

The world is constantly in motion and objects, as well as their properties, are subject to change. Therefore, a static representation of the world is insufficient, as it does not allow for the possibility of changes occurring as a result of actions. Some of these actions may be initiated by the agent by choice, for example, whereas others may be caused by external factors that the agent cannot control but only perceive and react to. We might then complement (1) and (2) above, with (3):

3. The theory should enable the modeling of actions, effects, and sensors, even those that are unreliable with a probabilistic model. It should also provide a mechanism for reasoning about how all of these entities impact the way knowledge about the world is evolving.

The above desiderata is virtually ignored in probabilistic modelling languages, at least in a general way that permits a comprehensive account of actions and sensing, and its complications such as the frame and ramification problem [43]. Indeed, probabilistic graphical models [49] and Kalman filters [56] are used for temporal phenomena, but under strict assumptions about how distributions are affected to capture a changing world. Although decision-theoretic [30] and probabilistic-planning languages allow the modelling of actions and effects [59], they are neither logics (in allowing for arbitrary connectives and quantifiers) nor general models of actions in terms of being able to reason about past and future histories. Relational probabilistic models [57], including dynamic ones [17], offer some logical features (such as clausal reasoning), but are not usually embedded in a theory of action and so do not provide a framework to reason about unbounded sequences of actions. This is not to say that such a framework could not designed starting from one of the more practical options – and indeed, there are many that come close [48] – but just that the search for a general and restriction-free option is still ongoing.

It is also not the case that these problems are completely solved by the knowledge representation community either. Consider, for example, that Reiter's [52] reconsideration of the situation calculus has proven enormously useful for the design of logical agents, essentially paving the way for cognitive robotics [34]. Among other things, it incorporates a simple monotonic solution to the frame problem, leading Reiter to define the notion of regression for basic action theories. The situation calculus has enjoyed numerous extensions for time, processes, concurrency, exogenity, reactivity, sensing and knowledge. Nevertheless, one criticism levelled at this line of work, and indeed at much of the work in cognitive robotics and reasoning about actions, is that the theory is far removed from the kind of probabilistic uncertainty and noise seen in typical robotic applications [56] and machine learning [46]. In fact, for many years, this criticism applied broadly to almost every knowledge representation language for reasoning about actions including, dynamic epistemic logic [58], the fluent calculus [55], among others [5, 53].

As discussed above, many recent attempts do come close to our desiderata, but often fall short in addressing (1) and (3) fully. For example, [48] unifies probabilistic relational languages, (not necessarily finite) discrete and continuous distributions, and a model of time to capture change to object properties. But it does not support arbitrary first-order logic or a full model of actions and unbounded sequences of actions. Analogously, [54] is a dynamic version of [42] – a probabilistic relational model with open-world features – but it does not support all of first-order logic or even arbitrary connectives. It also restricts how we can reason about sequences of past and future histories.

1.3 The case for meta-beliefs

A final ingredient ignored by almost all statistical relational learning and machine learning frameworks is the lack of a construct that can reason about meta-beliefs [22]. In areas from game theory [1] to distributed systems [44] to computer science [26], epistemic notions play an important role because we can easily distinguish between what is known versus what is true. For instance, we might be wanting to say that the box is red, but the robot *believes* it is blue, until it comes close to it and senses the *true* color. The notion of knowledge becomes even more important in a distributed and multi-agent setting [7] where we need to reason about the beliefs of many agents at the same time.

Moreover, each agent may need to hold beliefs about other agents so that they can collaborate, communicate and/or compete in a systematic way. Epistemic notions may also be useful for explaining automated systems [32, 6].

1.4 Our results

In this talk, I will briefly review our recent results on revisiting the unification of first-order logic and probability theory. In particular, we discuss the development of languages that support arbitrary quantification, possibly infinitely many random variables, both discrete and continuous distributions, as well as programming languages built on top of such features to include recursion and branching control. All of these languages are epistemic logics, and so support reasoning about beliefs and meta-beliefs as a bonus. In particular, we refer to the following results:

1. A first-order logic for actions and continuous distributions [9], but with finitely many random variables. (This model, however, is not a very convenient approach to reason about meta-beliefs.)
2. Query rewriting results for that logic that reduce an action sequence to some formula about the initial (static) knowledge base [10]. We may instead update the knowledge base [11].
3. A programming model with recursion for that logic [13].
4. A reconstruction of the logic with infinitely many random variables but with discrete distributions [8]. This model, in contrast, is an epistemic modal logic and permits reasoning about meta-beliefs in the usual way.
5. A model of program abstraction based on this epistemic modal logic [28]. This allows one to abstract the stochasticity away so that probabilistic programs can be mapped onto ones that mention no probabilities at all.
6. A refinement of the modal logic with infinitely many random variables and both discrete and continuous distributions [41].

2 A brief overview of results

Although it would be impossible to provide all the details of these contributions, we will give a quick overview of the key ideas behind these results here.

2.1 Language

We are interested in a first-order language. That is, it includes a set of predicates, function symbols, equality, first-order quantifiers (both universal quantification and existential quantification) and the usual connectives for conjunction, disjunction and negation. It will be simpler to often assume a fixed domain of discourse, say the set of natural numbers: $\mathbb{N}$, which serves as a domain of quantification.

Of course, if we want the language to also capture continuous distributions, we will also need to allow the domain to include the set of reals. The idea is that when we reason about objects (such as places, things and people), we will be quantifying over the countable domain of discourse. And when we need to reason about real-valued quantities, we will be using variable maps to go over the real numbers. The most general introduction to this can be found in [41].

2.2 Beliefs

If we simply want to talk about probabilities and how they change, our language can remain "objective." That is, we would need to only write down probabilistic expressions in our language and reason about them, as seen in [21]. However, since our aim here is to capture the beliefs of the robot, we will need epistemic operators. In particular, we will consider a "degrees of belief" modal operator that can be used as follows:

- $B(mass < 20) > 0.9$;
- $B(mass \neq 0 \wedge mass \neq 1) > 0.8$; and
- $K(mass^2 > 5 \vee mass^2 > 6)$, where $K\alpha \doteq (B\alpha = 1)$.

The first formula says that the agent believes the mass of an object being less than 20 units (say, kilograms) is greater than 0.9. We are dealing with normalized probabilities here so probabilities range between 0 and 1 inclusive. The second formula is saying that the agent believes the mass of the object not being 0 and not being 1 is greater than 0.8. Finally, the third one is saying that the square of the mass of this object being greater than 5 or being greater than 6 is believed with a degree of 1. The last formula is interesting because although it provides a probability for the disjunction, it does not quite say what exactly is a degree of belief for each of the disjuncts. This is the power of providing a general way to combine logic and probability because it allows you to express and quantify over arbitrary first-order formulas without specifying how exactly the probabilities should be assigned to the (ground) atoms in those formulas. This then raises the question: what does the semantics look like?

2.3 Semantics

The semantics for epistemic logics is usually given in terms of possible worlds [22]. The idea being that the agent considers some worlds possible and is said to know things only when those things are true in all the worlds considered possible. Ordinarily, when dealing with degrees of belief [20], we would say that the worlds are also assigned weights. This allows us to then say that some formula is believed with a degree of belief r if and only if the sum of the weights of the worlds that satisfy this formula comes up to r.

How then do we generalize this idea to deal with first order formulae, as well as the assigning of probabilities to complex formulas without providing the probabilities of atoms, as seen in the example of the disjunction above? For starters, we would say that each world is a first-order structure that interprets the predicates and the functions over the constants and the domain of the discourse in the usual way [19]. But because there are uncountably many worlds, we need to ensure that distributions are defined so as to obey the usual axioms of probability theory. There are a number of different ways to do this. We could either provide an appropriate analogue to the classical concept of defining a measure over an uncountable space [41]. Or, we could set up the probability of formulas based on countable additivity and convergence of the weights of worlds [8].

Such a framework, however, only accounts for a single distribution over the formulas. But because we are permitting situations where there may not be a unique distribution for formulas, we define an epistemic state as a set of distributions.

Once the semantics is set up this way, we can reason about disjunctions with the appropriate interpretation. For example: $B(p \lor q) = r$ for $r \neq 0$ does not imply $B(p) = n$ for any n. Note that, of course, in each of the distributions in the epistemic state, $B(p)$ will be equal to some n, but there is no single value agreed upon by all distributions in an epistemic state where $B(p \lor q) \neq 0$. In other words, the only obvious consequence of $B(p \lor q) \neq 0$ is $B(\neg p \land \neg q) \neq 1$.

Moreover, with the same semantical setup, we will now be able to reason about meta-beliefs and meta-knowledge very much in the spirit of introspective epistemic logics [15]:

- $(B(\alpha) = r) \supset K(B(\alpha) = r)$;
- $(\neg K\alpha) \supset K(\neg K\alpha)$;
- $(\alpha \equiv \beta) \supset (B\alpha = B\beta)$;
- $(B(\alpha) = r) \land (B(\beta) = r') \land (B(\alpha \land \beta) = r'')$ implies $B(\alpha \lor \beta) = (r + r' - r'')$.

The first formula says that if the agent believes a formula with degree r, then it knows (degree = 1) that it believes this. The second says that not knowing something means that the agent knows what it does not know. The third formula says that if two formulas are logically equivalent then the degree of belief for these formulas has to be the same. Finally the last formula is about the additive rule of probability.

2.4 Actions

To extend the model to actions [8], we can consider a language where a subset of the function symbols are reserved for actions. These function symbols could be things like *move*(x) which might say that the agent should move to x, and *pick*(x) which says that the agent should pick up the object x, and so on. To include these actions in the language, we will introduce two modal operators: $[a]$ and $\Box$, such that if ϕ is a formula, then so are $[a]\phi$ and $\Box\phi$. The first one says that ϕ is true after doing a. The second one says that after every sequence of actions, ϕ is true.

To extend the semantics, now we have to think of every world as a tree of first-order structures, because the world would be interpreting what is true initially as well as what is true after sequences of actions. The agent language is actually based on the situation calculus [52], but recast in a modal logic [33].

2.5 Basic action theories

To capture an application domain, we define a set of axioms which describes how the domain works: what actions are available, what sensors are available and what effects these have on the world and on the agent's knowledge. For example, consider [28]:

- $\Sigma_0 = at(near) \lor at(midpos) \lor at(far)$;
- $\Box Poss(a) \equiv \exists l.a = goto(l) \land \neg at(l)$; and
- $\Box[a]at(l) \equiv a = goto(l) \lor at(l) \land \neg\exists l' a = goto(l')$.

This says that initially the robot might believe to be close to an object (say, a wall), or at mid-distance from the object, or far away. The second one says that doing a go-to action is only possible when the robot is not already at the location. And finally the third sentence captures a monotonic solution to the frame problem [39] and basically says that the robot is at a location either if it executed a go-to action previously to that particular location, or it was already at this location and it did not go to some other location. Notice the box operator for the second and the third sentence. These conditions – the precondition and the effect of actions – is something that we want the worlds to obey after every action. So these are axioms that hold for all possible action histories.

A programming language can be defined on such axiomisations for high-level control [36]. For example, a program to get close to the wall and move away from the wall could be:

if $\neg At(near)$ **then** $goto(near)$ **endif**; $goto(far)$

Classically, most action languages have made a number of serious assumptions: noiseless sensors without measurement error and exact actions that always have the desired effect. In practice, both assumptions are idealistic: The sonar sensor may measure with an error, e.g., ± 1, or even have a continuous noise profile [56]. The robot may get stuck or fail with some probability. Most seriously, the robot may not be able to observe those errors.

2.6 Revisiting the axiomatisation

To address these assumptions, we need to revisit how we capture the domain. It may be too simplistic to simply say we can go to some location as an atomic action. And to say that we are either close or far from the wall might be too coarse to work with for a probabilistic representation. So we might end up using the following symbols in our language to capture the domain:

- $loc(x)$ is true if the distance to the wall is x;
- An action $move(x, y)$, where
  - x is the distance the robot intends to move;
  - y is the distance that the robot actually moves;
  - but y may not be observed by the robot; and
  - so, we will write $move(x)$ for $\pi y.\ move(x, y)$.

Here, πy is a nondeterministic pick operator for programs, the idea being that nature nondeterministically chooses y. Analogously, if $sonar(x)$ measures the distance to the wall and returns x, we let $sonar()$ be short for $\pi x.\ sonar(x)$ (where nature chooses x). Let us also assume that if a move by x units is intended, then the actual outcome is within one unit of the intended value. Likewise, whenever the sensor reads x, the true value is within one unit of that number. The probabilities accorded to these alternatives does not really matter for our informal discussion below, but for the sake of concreteness, say that each of $x - 1, x, x + 1$ taken on a probability of 1/3. (That is, the possible values are taken from a discrete uniform distribution.) Note that, by way of the language, we do not have to commit to a single distribution for the outcomes, much like how we did not need to commit to a single distribution for the robot's initial knowledge.

2.7 A revised program

With all of this machinery, the simple program from above, understandably, gets a lot more complicated and would look something like this:

```
$sonar()$
while $\neg K(\exists x.loc(x) \land x \leq 2)$ do $move(-1); sonar()$ end while
while $\neg K(\exists x.loc(x) \land x \geq 5)$ do $move(1); sonar()$ end while
```

So, while the robot does not know that it is close, move towards the wall (using a negative argument). But because the move action is noisy, the robot needs to sense. And the sense-act loop needs to repeat until the robot has full certainty that it is actually close. It then exits the first while-loop and enters the second while-loop. Here, as long as it does not believe that is far, it moves away from the wall and senses to counter the noisy action. This repeats until it is actually away from the wall.

3 Abstraction at play

Probabilistic belief programs seem to capture the intuitions perfectly, and they corresponding to cyclic policies [35], but they present several challenges, at least when written down by hand. Firstly, correctly designing these programs is a complex task. The incorporation of probabilities into the program structure requires a careful assessment of how sensed values, which are not in the robot's control, need to determine what to do next. Secondly, reasoning about probabilities can be a difficult endeavor for the modeler. It is challenging to accurately assess the likelihood of different outcomes or events, especially over long histories. Lastly, comprehending how and why a probabilistic belief program works is nontrivial.

So, ideally, we would like to write high-level programs without having to deal with probabilities. And, by doing so, one can obtain models and execution traces that are easy to understand. This is where abstraction comes in. The (low-level) action theory includes stochastic actions as usual, but then a high-level action theory is defined that abstracts away stochasticity [28]. For example, it is possible to show that there is a mapping for predicates and actions for the numeric example above such that the above programming involving two while-loops has the same behavior as the one with if-then-endif (the first program introduced above)! This is because the goto action in the if-then-endif program maps to a sense-act while-loop for the low-level action theory, allowing us to establish that formally the two programs are equivalent for a particular mapping. Obviously, this has a huge advantage because it is very easy to design the first program. It almost corresponds to how we might describe the instructions in natural language. In contrast, the second program is much harder and requires an understanding of probability theory and dynamics.

3.1 A note about continuous noise

The discussion above was largely based on a discrete model of probability theory embedded in first-order logic. This is why we were able to consider the degree of belief taking a value of exactly one. The sensor model, moreover, also introduced an error within one unit of the actual value. In reality, in continuous models, this becomes far more complex. Beliefs about the initial situation may take on uncountably many values. Both the sensor and effector can have, say, Gaussian error profiles [56]. What this means is that we still require sense-act loops, but it is highly unlikely that the degree of belief in a formula of interest would be exactly one. We might have to contend with a degree of belief greater than, say, 0.9 as conditions in the programs. It would even make sense to involve the expected values of the probabilistic variables, as discussed in [13].

4 Future work

Much of this discussion has largely been on a theoretical front in terms of searching for general-purpose languages where restrictions are not placed purely for computational reasons. We are interested in developing the semantics and the mathematical theory that allows us to reason about properties in languages that allow a full fragment of first-order logic + all of probability. As argued, this is a much needed mathematical exercise, but in reality we would want to work with robots and virtual agents. So the obvious question here is how can we make any of this tractable, scalable or efficient?

To do that, we need to revisit areas such as statistical relation learning and neurosymbolic AI, which have achieved considerable success in developing concise languages [51]. These languages not only capture interesting probability distributions described over symbolic artifacts but also provide implementations and, in some cases, tractability guarantees [16]. However, as we have argued, these areas do not offer a complete understanding of how probability theory integrates with first-order logic over a general theory of actions. Hence, our above exercise sets the stage. With that in hand, we may now strive to find a middle ground, where we can explore fragments that are slightly more expressive than existing proposals in statistical relation learning. These fragments should retain certain features mentioned earlier, including the ability to define recursive programs and reason about hypothetical events and future histories, much like in temporal logic [18].

To some extent we have some early results that do exactly this. For example, in [12], we showed how reasoning about actions can be reduced to a formula about the initial situation, after which we can use any existing reasoning module for the initial situation, including Bayesian networks and relational probabilistic models. Interestingly, this type of reduction strategy was explored independently for the practical problem of robot task and motion planning [31]. In [13], we showed that the kinds of belief-based programs discussed above, involving degrees of beliefs as conditions in the program, can be implemented using a particle-filtering strategy provided the language was appropriately restricted. By extension, it would be interesting to also develop fragments where it is possible to combine discrete and continuous distributions in natural way, and reason about things like abstraction. There is some recent work on verifying such programs [40].

It might be also possible to connect these advances in knowledge representation to exciting work in ML on probabilistic programming [24], where the topic of synthesis and learning are also embraced. For example, there is already work on synthesizing (recursion-free) probabilistic program abstractions [29] and perhaps it is possible to leverage such ideas for the programs discussed above.

More broadly, despite the advances of deep learning, it is now becoming increasingly clear that symbolic artifacts play an important role from the perspective of explainability, domain adaptation, domain knowledge inclusion, as well as reasoning. Neuro-symbolic AI is a representative example for the scientific need for this type of unification [27]. However, ad-hoc or otherwise limited combinations of logic and probability are likely to only take us so far: we need to think more carefully about how general-purpose open-ended agents might reason with quantifiers and how they might reason about past histories and future outcomes. All of these latter features is precisely what motivates many expressive knowledge representation languages. By embracing probability theory, we are now in a position to draw concrete connections between expressive KR languages and learning, and offer capabilities, including reasoning about meta-beliefs and belief-based program verification, that are out of reach for standard machine learning theory based on probability theory alone.

References

1. R. J. Aumann. Interactive epistemology II: probability. *Int. J. Game Theory*, 28(3):301–314, 1999.
2. F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, 1990.
3. F. Bacchus, J. Y. Halpern, and H. J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111(1–2):171 – 208, 1999.
4. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.
5. C. Baral and M. Gelfond. Logic programming and reasoning about actions. In *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 389–426. Elsevier, 2005.
6. V. Belle. Counterfactual explanations as plans. In *The 39th International Conference on Logic Programming*. Open Publishing Association, 2023.
7. V. Belle, T. Bolander, A. Herzig, and B. Nebel. Epistemic planning: Perspectives on the special issue, 2022.
8. V. Belle and G. Lakemeyer. Reasoning about probabilities in unbounded first-order dynamical domains. In *IJCAI*, 2017.
9. V. Belle and H. J. Levesque. Reasoning about continuous uncertainty in the situation calculus. In *Proc. IJCAI*, 2013.
10. V. Belle and H. J. Levesque. Reasoning about probabilities in dynamic systems using goal regression. In *Proc. UAI*, 2013.
11. V. Belle and H. J. Levesque. How to progress beliefs in continuous domains. In *Proc. KR*, 2014.
12. V. Belle and H. J. Levesque. PREGO: An Action Language for Belief-Based Cognitive Robotics in Continuous Domains. In *Proc. AAAI*, 2014.
13. V. Belle and H. J. Levesque. Allegro: Belief-based programming in stochastic dynamical domains. In *IJCAI*, 2015.
14. R. Carnap. *Logical foundations of probability*. Routledge and Kegan Paul London, 1951.
15. B. Chellas. *Modal logic*. Cambridge University Press, 1980.
16. A. Choi and A. Darwiche. On relaxing determinism in arithmetic circuits. *arXiv preprint arXiv:1708.06846*, 2017.

17. J. Choi, A. Guzman-Rivera, and E. Amir. Lifted relational kalman filtering. In *Proc. IJCAI*, pages 2092–2099, 2011.
18. C. Dixon, M. Fisher, and B. Konev. Tractable temporal reasoning. In *IJCAI*, volume 7, pages 318–323, 2007.
19. H. Enderton. *A mathematical introduction to logic*. Academic press New York, 1972.
20. R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.
21. R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and computation*, 87(1-2):78–128, 1990.
22. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
23. H. Gaifman. Concerning measures in first order calculi. *Israel Journal of Mathematics*, 2(1):1–18, 1964.
24. A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani. Probabilistic programming. In *Proc. International Conference on Software Engineering*, 2014.
25. J. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, 1990.
26. J. Y. Halpern, R. Pass, and V. Raman. An epistemic characterization of zero knowledge. In *TARK*, pages 156–165, 2009.
27. P. Hitzler. Neuro-symbolic artificial intelligence: The state of the art. 2022.
28. T. Hofmann and V. Belle. Abstracting noisy robot programs. *AAMAS*, 2023.
29. S. Holtzen, G. Broeck, and T. Millstein. Sound abstraction and decomposition of probabilistic programs. In *International Conference on Machine Learning*, pages 1999–2008. PMLR, 2018.
30. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99 – 134, 1998.
31. L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *I. J. Robotic Res.*, 32(9-10):1194–1227, 2013.
32. S. Kambhampati. Challenges of human-aware ai systems. *AI Magazine*, 41(3), 2020.
33. G. Lakemeyer and H. J. Levesque. Situations, Si! situation terms, No! In *Proc. KR*, pages 516–526, 2004.
34. G. Lakemeyer and H. J. Levesque. Cognitive robotics. In *Handbook of Knowledge Representation*, pages 869–886. Elsevier, 2007.
35. J. Lang and B. Zanuttini. Knowledge-based programs as plans - the complexity of plan verification. In *Proc. ECAI*, pages 504–509, 2012.
36. H. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84, 1997.
37. H. J. Levesque. *Common sense, the Turing test, and the quest for real AI.* Mit Press, 2017.
38. H. J. Levesque and G. Lakemeyer. *The logic of knowledge bases*. The MIT Press, 2001.
39. H. J. Levesque, F. Pirri, and R. Reiter. Foundations for the situation calculus. *Electron. Trans. Artif. Intell.*, 2:159–178, 1998.
40. D. Liu. *Projection in a Probabilistic Epistemic Logic and Its Application to Belief-based Program Verification*. PhD thesis, RWTH Aachen University, 2023.
41. D. Liu, Q. Feng, V. Belle, and G. Lakemeyer. Concerning measures in a first-order logic with actions and meta-beliefs. In *20th International Conference on Principles of Knowledge Representation and Reasoning*, 2023.
42. B. Milch, B. Marthi, S. J. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *Proc. IJCAI*, pages 1352–1359, 2005.
43. L. Morgenstern and S. A. McIlraith. John McCarthy's legacy. *Artificial Intelligence*, 175(1):1 – 24, 2011.

44. Y. Moses, D. Dolev, and J. Y. Halpern. Cheating husbands and other stories: A case study of knowledge, action, and communication. *Distributed Computing*, 1(3):167–176, 1986.
45. S. Muggleton, L. De Raedt, D. Poole, I. Bratko, P. Flach, K. Inoue, and A. Srinivasan. Ilp turns 20. *Machine learning*, 86(1):3–23, 2012.
46. K. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.
47. N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.
48. D. Nitti. *Hybrid Probabilistic Logic Programming*. PhD thesis, KU Leuven, 2016.
49. J. Pearl. Graphical models for probabilistic and causal reasoning. In *Quantified Representation of Uncertainty and Imprecision*, pages 367–389. Springer, 1998.
50. D. Poole. Logic, probability and computation: Foundations and issues of statistical relational AI. In *Logic Programming and Nonmonotonic Reasoning*, volume 6645 of *LNCS*, pages 1–9. 2011.
51. L. D. Raedt, K. Kersting, S. Natarajan, and D. Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.
52. R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.
53. R. B. Scherl, T. C. Son, and C. Baral. State-based regression with sensing and knowledge. *Int. J. Software and Informatics*, 3(1):3–30, 2009.
54. S. Srivastava, S. J. Russell, P. Ruan, and X. Cheng. First-order open-universe pomdps. In *UAI*, pages 742–751, 2014.
55. M. Thielscher. From situation calculus to fluent calculus: state update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111(1-2):277–299, 1999.
56. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
57. G. Van den Broeck, W. Meert, and J. Davis. Lifted generative parameter learning. In *Statistical Relational Artificial Intelligence, AAAI Workshop*, 2013.
58. H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007.
59. S. W. Yoon, A. Fern, and R. Givan. Ff-replan: A baseline for probabilistic planning. In *ICAPS*, pages 352–, 2007.