On the relevance of logic for AI, and the promise of neuro-symbolic learning

Vaishak BELLE [a]

[a] *University of Edinburgh & Alan Turing Institute, UK*

Abstract

In this position paper, we examine some of the assumptions held about logic and its relevance to the development of modern AI, which is primarily driven by deep learning. The paper aims to address fundamental misunderstandings about logic and ultimately argue for the benefits of symbolic formalisms in modeling uncertain worlds. While it is now recognized that statistical associations learned from data are limited in their ability to understand the world, there is still a great deal of criticism and hesitancy regarding the use of symbolic logic to achieve or support a broader vision for AI. By arguing that symbolic logic is more flexible than non-experts believe, we make a case for Neuro-Symbolic AI, which offers the best of both worlds.

Keywords. Neuro-symbolic AI
Logic and learning

Preface

This article lies between a position paper and a survey paper, and it does two things. On one hand, it discusses the breadth and diversity of solutions encompassed within symbolic logic. We believe that many of these dimensions are not obvious to people outside the logical community, and perhaps, even those working within certain areas of logic might not be aware of the latest developments in statistical relational learning. On the other hand, it points to common objections to using logic when building complex AI systems involving machine learning. Essentially, this paper serves as a survey that tackles both of these aspects.

1. Introduction

Artificial Intelligence (AI) is widely acknowledged as a new kind of science that will bring about (and is already enabling) the next technological revolution. Virtually every week, exciting reports come our way about the use of AI for drug discovery, game playing, stock trading and law enforcement. And virtually all of these are mostly concerned with a very narrow technological capability, that of predicting future instances based on past instances.

Identifying statistical patterns, correlations, and associations is, without doubt, extremely useful. In the first instance, it is needed in numerous applications to inspect features and properties of interest in observed data. It serves as the backbone of recommendation systems, for example, and is likely more than sufficient, even with flaws, when gathering context. While searching for “how to raise lambs” in an online bookstore, we might be a little disappointed if it suggests “Silence of the Lambs” by Thomas Harris, and somewhat annoyed if it suggests cookbooks on “how to cook lamb”, but such low quality results are unlikely to have long-term effects. This type of AI might also be useful but somewhat problematic for, say, fast-tracking the review of job applications, provided these models are adjusted for bias, and a human intervenes and interprets the outcome and determines how to act further. This type of AI was largely believed to be sufficient for vision systems [185], until it was observed that self-driving cars fail stupendously, and that the state-of-the-art systems can be fooled in strange and unnatural ways [79].

Be that as it may, this is a very narrow view of AI capabilities. AI, as understood by both scientists and science fiction writers, is clearly much broader. What distinguishes big-data analysis from AI is that the set of capabilities we wish to enable with the latter. We are interested in a general-purpose, autonomous computational entity that, in the very least, has agency. Many of these concerns were widely debated, discussed, and developed during the heyday of good old-fashioned AI [55,121,116].

However, despite recognizing that data-driven statistical learning is limited in its ability to understand the world and model its knowledge [130], there is still a lot of criticism and hesitancy about the use of symbolic logic to accomplish or assist in a broader vision for AI [48].

In this position paper, we examine some of the assumptions held about logic and its relevance to the development of modern AI, which is primarily driven by deep learning. The paper aims to address fundamental misunderstandings about logic and ultimately argue for the benefits of symbolic formalisms in modeling uncertain worlds. By arguing that symbolic logic is more flexible than non-experts and critics believe, we make a case for Neuro-Symbolic AI, which offers the best of both worlds.

2. Logic is old-fashioned

In the first part of this article, we will look at some of the criticisms against using logic. We then turn to a number of positive dimensions to examine the integration of logic and learning.

2.1. Neural approaches and nothing else!

Modern AI has moved on, we are told. The idea of using symbolic logic is outdated, and the area of knowledge representation defined over symbolic logic is now affectionately (or perhaps pejoratively) called good old-fashioned AI, or GOFAI for short. For example, in [28], Bengio et al. write:

> ... *machine learning is the only viable approach to building AI systems that can operate in complicated real-world environments.*

In the early days of AI, John McCarthy put forward a profound idea to realise artificial intelligence (AI) systems [131]: he posited that what the system needs to know could be represented in a formal language, and a general-purpose algorithm would then conclude the necessary actions needed to solve the problem at hand. The main advantage is that the representation can be scrutinised and understood by external observers, and the system's behaviour could be improved by making statements to it.

Numerous such languages emerged in the years to follow, but first-order logic remained at the forefront as a general and powerful option [141]. Propositional and first-order logic continue to serve as the underlying language for several areas in AI, including constraint satisfaction [30], automated planning [165], database theory [123], ontology specification [109], verification [11], and knowledge representation [121].

And yet, "modern" AI has decided that these efforts are superfluous, or at least easily replaceable once a training dataset has been created. For example, Sejnowski [172] writes:

The early goals of machine learning were more modest than those of AI. Rather than aiming directly at general intelligence, machine learning started by attacking practical problems in perception, language, motor control, prediction, and inference using learning from data as the primary tool. In contrast, early attempts in AI were characterized by low-dimensional algorithms that were handcrafted. However, this approach only worked for well-controlled environments. For example, in blocks world all objects were rectangular solids, identically painted and in an environment with fixed lighting. These algorithms did not scale up to vision in the real world, where objects have complex shapes, a wide range of reflectances, and lighting conditions are uncontrolled. The real world is high-dimensional and there may not be any low-dimensional model that can be fit to it. Similar problems were encountered with early models of natural languages based on symbols and syntax, which ignored the complexities of semantics. Practical natural language applications became possible once the complexity of deep learning language models approached the complexity of the real world. Models of natural language with millions of parameters and trained with millions of labeled examples are now used routinely.

He goes on to suggest that:

Is there a path from the current state of the art in deep learning to artificial general intelligence? From the perspective of evolution, most animals can solve problems needed to survive in their niches, but general abstract reasoning emerged more recently in the human lineage. However, we are not very good at it and need long training to achieve the ability to reason logically. This is because we are using brain systems to simulate logical steps that have not been optimized for logic. Students in grade school work for years to master simple arithmetic, effectively emulating a digital computer with a 1-s clock. Nonetheless, reasoning in humans is proof of principle that it should be possible to evolve large-scale systems of deep learning networks for rational planning and decision making.

The "theory of everything" approach in science, or perhaps its analog in AI, that of having a single algorithm/architecture/framework for all tasks [59], is undoubtedly appealing. Some theoretical physicists have hopes pinned on string theory, for example, to come up with a single framework that unifies all observational data, across large and minuscule physical bodies [57]. Likewise, the appeal of purely neural model is attractive. However, there is lots to debate here.

Firstly, deep learning models are loosely inspired by the brains but not fully accurate representations (yet) [180,139].

Secondly, there is the notion of innateness [186], and how much evolution might help the brain in understanding and processing the world in a structured manner. And thirdly, we must bear in mind that we still lack a complete understanding of how the neurons of a bird (let alone a human) are wired, and how that influences cognitive capabilities. Merely knowing that neural weights enable birds to solve puzzles and recognize faces does not necessarily imply that our implementation of their neurons should resemble or possess similar properties. These concerns had also been debated in the literature in the 1980s [162,177].

Lastly, Sejnowski [172] also offers the social element of learning:

Although the focus today on deep learning was inspired by the cerebral cortex, a much wider range of architectures is needed to control movements and vital functions. Subcortical parts of mammalian brains essential for survival can be found in all vertebrates, including the basal ganglia that are responsible for reinforcement learning and the cerebellum, which provides the brain with forward models of motor commands. Humans are hypersocial, with extensive cortical and subcortical neural circuits to support complex social interactions.

Putting such issues aside, it is also worth noting that proponents of the symbolic approach to AI never explicitly claimed the existence of symbolic representations within our minds [32,121]. In essence, the symbolic approach offers a coherent strategy for: (a) executing symbolic expressions, which capture the knowledge of the system about the world, and (b) comprehending the (idealized) implications of one's knowledge, as specified by inference rules in logic.

As argued by Levesque [119], this is not a novel concept – Leibniz articulated centuries ago that certain types of thinking adhere to symbolic processing. Hence, why not employ an algebraic treatment for cognition? As scientists, we may debate whether it is more useful to have an exact model of computation that approximates the reasoning in the brain [177,99] or whether we should forego these models altogether and simply be satisfied with informal descriptions of reasoning [156], as might emerge from a trained model [47].

We reiterate that the allure of a purely neural approach is understandable, given its simplicity and the sense of a "unified theory" it evokes. However, the arguments regarding the effectiveness of the training process in capturing intricate reasoning [91] and the potential for incorrect [189] and unreliable predictions [5] suggest that a purely neural approach may not be sufficiently robust to exploit and capture structure. Indeed, despite Sejnowski [172] offering that "it should be possible to evolve large-scale systems of deep learning networks for rational planning and decision making", he admits also that "a hybrid solution might also be possible, similar to neural Turing machines developed by DeepMind for learning how to copy, sort, and navigate."

More generally, by taking a step back, we realize that until the past few centuries, our understanding of the brain and neurons was limited. Yet, during this time, we were able to calculate, develop number theory, construct calculators, and ultimately build computers [187]. Imagine if we had solely dedicated ourselves to constructing elaborate brain replicas in the hopes that they could handle (say) tax calculations for us. Most importantly, we cannot test for a capability without first defining that capability, such as (say) deduction [156].

All of this underscores the significance of the symbolic approach, which offers an idealized framework for well-defined (relative to the formal language) forms of reasoning. There is a popular analogy [32] suggesting that we need not build wings and feathers to build airplanes; comprehending the principles of aerodynamics is enough. So, why shouldn't the development of a theory of artificial cognition be just as relevant for a type of AI that is behaviorally similar to humans in some instances, without necessarily resorting to a brain-like architecture? Or perhaps a combination of the symbolic and the neural, as offered by neuro-symbolic AI [90]? The mistrust may stem from the misconception that logic and probabilistic learning are fundamentally incompatible or entirely separate domains—a notion we will now challenge.

2.2. *There is a dichotomy*

A common view held by many in the broader community that there is an inherent dichotomy between symbolic logic and machine learning — the former for discrete domains, and the latter for continuous ones. The exact boundary between "discreteness" and symbolic logic might be obfuscated even in works that are strong proponents of using logic in machine learning. For example, in one of the most popular representations of probabilistic relational learning — so-called Markov logic networks (MLNs) — the following is said:

- *First-order logic (with Assumptions above) is the special case of MLNs obtained when all weights are equal*
- *Every probability distribution over discrete or finite- precision numeric variables can be represented as a Markov logic network.*

The assumptions these statements refer to ensure that the set of constants in the domain of discourse is finite, leading to finitely many possible interpretations, all of which are of finite size. A reader not familiar with logic might incorrectly infer that we are only able to construct discrete probability distributions using first-order logic. Likewise, even in a nuanced survey such as [38] on the importance of unifying logic and learning, they write:

- *The term 'discrete representation', used throughout this paper, denotes a discretely valued variable that represents some concept, which can take on either a limited or a countably infinite number of distinct values.*
- *Discrete processing consists of the application of any discrete function to input data. A discrete mathematical function has a domain, and hence a range, consisting only of discrete sets of values. Examples can be found in integer arithmetic, in computer programming languages and in first order logic.*
- *As deep models are currently trained using gradient descent, it is relevant to note that discrete functions are not differentiable (in any subset of their domain).*
- *Modern neural networks are representative of continuous processing. First order logic or symbolic AI models are representative of discrete processing.*

Although they go on to mention fuzzy logic and non-monotonic logics later, readers might still come away with the impression that symbolic logic is primarily suited for discrete entities. To be clear, first-order logic, interpreted over a finite or a countably infinite domain and interpreted classically, does not lend itself to differentiability. We will return precisely to this point, but non-logicians might conclude: (a) symbolic logic as used in AI is focused on discrete symbols; and (b) symbolic processing in vector (real-valued) space is a separate topic of study that can be independently done from symbolic logic.

What we are seeing here is a narrowing of the use of "logic" simply as *classical logic* – say, as introduced in [63] – defined over Boolean truth values. Moreover, the use of logic is also assumed to be limited to discrete propositional assertions, as seen in ontologies that capture relationships and hierarchies about commonsensical concepts [132], as well as in early attempts at logic programming [110].

We will now discuss the use of non-Boolean truth values and continuous properties in logic, and how that is making an appearance in the area of neuro-symbolic AI. The subtlety and clarification here is that indeed logical objects are discrete entities, or more precisely, discrete structures. To compute entailments, moreover, we algebraically manipulate symbols. However, these structures might capture continuous properties either by allowing non-binary truth values, by using function symbols over the real space, or by distributions on the models of formulas. This leads to various paradigms of relational learning and neuro-symbolic AI, many of which are differentiable.

2.2.1. Real-valued truth values

To a large extent, it is true that the area of knowledge representation in AI focuses on discrete symbols and a Boolean interpretation [32]. But, on the other hand, it's been close to 60 years since we have fuzzy logic [206], among others languages for non-binary truth values [105]. These allow us to assign a truth value between 0 and 1 to propositions, with the understanding that these values indicate the degree to which the proposition may be true. Fuzzy logic can also be utilized to represent ambiguous concepts, such as stating that a person is tall, without specifying tall as a categorical property.

The use of such values to propositions means that the interpretation of Boolean connectives also changes. For example, the formula $\alpha \wedge \beta$ could be mapped onto the *min* of the truth values of the individual formulas. That is:

$$\alpha \wedge \beta \doteq min(\alpha, \beta).$$

If α is assigned a truth value of 0.6 and β is assigned a truth value of 0.4, then the conjunction would be given a truth value of 0.4. Of course, one can see that if the truth values are either 0 or 1, then the *min* function aligns with classical logic in the sense that if either α or β is 0, then the conjunction will also have the truth value of 0.

By construction, the outputs of neural networks can be mapped to real numbers between 0 and 1. Owing to the nature of truth values in such logics, these outputs can be directly modeled as atoms in logical formulas. This led to an early wave of *neurosymbolic AI* formalisms [74] and the development of a field that integrates neural outputs in a logical language [90]. Perhaps the most representative examples in this space are logic tensor networks [8] and other approaches based on fuzzy logic [195]. The motivation for many of these languages is to logically capture concepts that have been learned from neural networks, in order to reason about these concepts as part of a commonsensical knowledge base. Thus, the agent would be reasoning about hierarchies and relationships, but many of the relations in this knowledge are learned directly using neural networks, presumably from observational data.

It is worth noting that reasoning about concepts and relations is an ongoing problem with neural networks – see efforts such as capsule networks [169] and module networks [4] – and there are very few general solutions. Neuro-symbolic AI is stepping in here, especially if it were to allow a general framework for injecting knowledge expressed in a fragment of first-order logic, could be very welcome.

2.2.2. *From discrete to continuous*

Capturing the output of neural networks as truth-values in a logical formula is one approach to reasoning about vector spaces. However, we can also use logic to reason about continuous properties as formulas.

Although it is common to discuss discrete properties in logical AI, it is not necessary that they must do so. Logical formulas are indeed discrete structures, but they can also express properties about countably infinite or even uncountably many objects [88,161, 26,17].

Reasoning about real numbers have long been an area of interest in mathematical logic [100], going back to Tarksi, and are a major concern in satisfiability modulo theories (SMT) [11]. SMT can be seen as a generalization of SAT for propositional logic and is being used for the verification of timed and hybrid systems that involve both discrete and continuous properties. For example, the following formula expresses that a logical function with one argument f applied to x is lesser than the square of y. This could be conjoined using Boolean connectives with other assertions, such as one that says y is greater than the two-variable function g applied to x and z:

$$f(x) \leq y^2 \wedge y > g(x, z).$$

Here, the domain of $x, y,$ and z could range over the set of natural numbers $\mathbb{N}$, the set of integers $\mathbb{Z}$, or even the set of reals $\mathbb{R}$.

Therefore, we can use these formulas to represent constraints on geometric spaces. A recent body of work has examined the idea of regularizing neural networks by adding logical constraints to the loss functions. The idea is to train the network such that the loss is calculated against this logical constraint, which is backpropagated. The goal then is to train the network in such a way that predictions always satisfy these logical constraints. There is existing work on propositional constraints [73], real-valued constraints [91] as well as temporal formulas [97], the latter of which trains the network to dynamically navigate an environment in only the valid geometric space.

One of the interesting observations in almost all of these papers on loss functions is that they demonstrate that it is much more effective to train the network using such loss functions than assuming the constraints are represented in the data. So it is much more sample efficient [96]. Moreover, some of these architectures also allow for the complete satisfaction of the constraints [91]. This is necessary in safety-critical and high-stakes applications.

2.3. *Logic is not good for probabilistic uncertainty*

Classical quantifiers in logic, as well as the connectives, allow for disjunctive uncertainty, the existence of individuals, and properties applicable to all individuals in the domain. Because the data we collect is often noisy, or we sometimes have to approximate and average over populations, the use of probability theory is essential [150]. Since classical logic traditionally did not represent probabilistic assertions, much of the learning and uncertainty in the AI community moved away from logic. We will discuss here that the connection between logic and probability is deep, and there is a vibrant community focussed precisely on this agenda [159].

2.3.1. Probabilistic logical models

Since the work of Nilsson [143], the use of logic to capture non-trivial probabilistic spaces and reason logically about events in those spaces has been a major concern in uncertainty quantification in AI [168] and statistical relational learning [159]. The key idea here is that it should be possible to assign probabilities to atoms, which would then provide a way to extend these probabilities to complex formulas. That is, if α is a well-defined (classical) formula in a logical language $\mathcal{L}$, then so is $\Pr(\alpha)$ [84]. This leads to a representation language that may involve a combination of deterministic and probabilistic assertions, capturing the knowledge base of a putative agent. For example [25], consider the following formula:

$$\alpha \wedge \Pr(\beta) > \Pr(\gamma) \wedge \Pr(\gamma) \leq 0.6.$$

It is assumed that α is true, and the probability that β is true is greater than the probability that γ is true. Additionally, γ is believed with a probability of less than or equal to 0.6. Here, α may be a non-probabilistic assertion. The probability of γ is not given a unique value, and we are allowed to compare the likelihoods of two formulas. Such combinations are difficult to express using probability theory alone.

In recent years, there has been a steady progress on designing languages that can not only capture Bayesian networks and factor graphs [113], but also extend them with a relational and a logical syntax. Popular languages for pragmatic specifications of logic and probability include Markov logic networks [164], ProbLog [160] and BLOG [136]. Many of these not only investigate the representational restrictions that enable the capture of distributions succinctly, but also explore how to reason with the resulting distribution, and in some cases, learn the distributions or representations themselves. (They have to restrict the expressiveness of the language in order to ensure that their representations capture a single distribution; so the above formula may be difficult to express here too.) Consider the following program in ProbLog [160]:

```
0.5::heads1.
0.6::heads2.
twoHeads :- heads1, heads2.
```

This allows us to capture a mixture distribution composed of a biased coin toss and an unbiased coin toss, with the latter having a 0.6 probability of landing heads.

Interestingly, Bayesian networks can also be modeled as ProbLog programs [159]. And what is more interesting is that probabilistic inference in Bayesian networks [42], ProbLog programs [69], Markov logic networks [164], and factor graphs [113] can all be shown to be reducible to the same computational task known as *weighted model counting* [6]. Weighted model counting is an extension to SAT in the sense that each satisfying assignment is assigned a weight. By computing the sum of the weights of all satisfying assignments, we can relate that sum to the conditional probability and marginals in a Bayesian network. That is, for a propositional language $\mathcal{L}$, assume a weight function w maps its literals to $\mathbb{R}_{[0,1]}$. Then, for some $\phi \in \mathcal{L}$,

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

The product operation here is defined in terms of all the literals that are true in a given model of ϕ.

As argued in [193,16], it is not only the case that logical languages allow us to reason about probability distributions over combinatorial spaces, but it is also the case that the syntax of logic can help capture complex relationships that are difficult to model using standard probabilistic languages [77]. Moreover, by way of weighted model counting, there is a single generic approach for probabilistic reasoning over discrete, combinatorial spaces that is competitive [42]. It is also amenable to both exact as well as approximate inference schemes [40].

Recently, there have also been extensions from discrete combinatorial spaces to continuous ones [27,44], referred to as *weighted model integration*. Here, the formula $x \in [-5, 5]$ with a weight of 0.56 might represent a continuous random variable x whose piecewise constant density for all values between -5 and 5 is 0.56. Analogously, the same formula with the weight of $x^2/2$ might represent a piecewise polynomial density specification for x, such that for all values between -5 and 5, its density is given by the square of that value divided by 2. As with weighted model counting, inference in this formulation is performed by means of a notion of model counting in SMT [11].

2.3.2. Generalising the specification of a distribution

Going back to the history of the use of logic in AI [141], there has been considerable interest in unifying logic and uncertainty. Note that, through the use of quantifiers, it is possible to express uncertainty that may not always align with a single distribution. For instance, McCarthy [133] was concerned about probabilities in the early years of using first-order logic for knowledge representation. However, he makes a very salient point that we need to think carefully how numbers and first-order sentences fit together. For example, he argues [133]:

(i) It is not clear how to attach probabilities to statements containing quantifiers in a way that corresponds to the amount of conviction people have.

(ii) The information necessary to assign numerical probabilities is not ordinarily available. Therefore, a formalism that required numerical probabilities would be epistemologically inadequate.

His point, simply, is that we should not be expected to put probabilities on every formula; sometimes it suffices to say that $p \vee q$ holds without saying which, and by how much. Moreover, if we assign a probability of r on that formula, or to, say, $\exists x P(x)$, such an assertion in itself does not provide any additional information on how to further assign a probability to p, q, $P(a)$, and so on. Many popular languages for logic and probability mentioned above, including Markov logic networks [164], ProbLog [160] and BLOG [136], do not allow this level of flexibility. In fact, this requires a different type of machinery altogether, one which permits multiple prior distributions [24]. Consider a sub-formula from the example from above:

$$\Pr(\gamma) \leq 0.6.$$

The formula should, in principle, allow for every distribution that accommodates a probability of γ being less than or equal to 0.6. In contrast, in ProbLog, it is assumed that there is a single distribution over the model, and not specifying a probability on (say) a disjunction might be interpreted as a hard constraint that is true in all possible worlds. However, there are languages that do permit such rich specifications. See, for example, works such as [146] and [25].

More generally, probability measures [72] on first-order structures and other proposals on logic and uncertainty [164,160,136,24] allow us to append probabilities and weights in a logical language in different ways, yielding formal frameworks that go beyond and generalize the standard definition for a probability space. There are also approaches [60] that are based on possibility theory, which permits a different model for uncertainty that can be powerful when experts disagree or are uncertain about probabilistic assertions.

2.4. Symbols without (explicit) semantics

In the machine learning literature, it is not uncommon to find syntactical objects, especially well-defined symbolic expressions, such as programs, that are learned without an explicit definition of the semantics [115]. In such cases, one would need to define only the interpreter and the compiler [62], with an implicit notion that the atomic objects refer to concrete objects in the real world, as obtained by the process of *symbol grounding* [183].

However, with programs in the program induction literature [82], there is (or rather, should be) an implicit logical syntax and semantics that defines: (a) what sort of expressions can be constructed, and (b) what they mean and capture. For example, sequential instructions could be understood as conjunctions, and while loops can be captured using second-order quantification [118,184,82]. If we further want to understand what properties are entailed by these programs, then we need to define the semantics comprehensively and analyze what follows from the logical theory corresponding to a program.

Indeed, without a clear specification of how compositions of expressions should be interpreted and evaluated, how are we to know what these programs are yielding [131]? There has been a surge of a new family of programming languages that capture intricate machine learning models. Typically, these languages allow the use of random primitives as well as operators for conditioning and providing evidence. These are referred to as *probabilistic programming* – see, for example, Church [80], ProbLog [53], and the generic construction in [178]. In some cases, they might support combinations of discrete and continuous distributions, and higher-order functions. A general approach to understanding how these programs can be constructed and what sort of distributions they model is through the use of a formal semantical setup, usually in a fragment of first or second-order logic.

See also works such as [12] for discussions on attempting to construct the semantics for one programming language syntax from another. Such a move is especially desirable if we want to check for the internal consistency of an ad hoc programming language. For philosophical arguments on the importance of semantics, see, for example, [46].

2.5. Logic is about categorical propositional assertions

As discussed above, often “logic” is synonymous with (the classical interpretation of) propositional logic.

There are many systems for writing down symbols, and interpreting logical symbols and formulas built up these symbols. Classical approaches include propositional logic (Boolean symbols, A and B is true iff A is true and B is true) and first-order logic, which uses quantifiers. In first-order logic, there is a domain of discourse which stands for the objects in the world. We then say that $\exists x.\ P(x)$ is true if and only if there is some individual from the domain of discourse such that the property P is true for that individual. Likewise, the formula $\forall x, y.\ \mathit{Grandparent}(x, y) \supset \exists z.\ \mathit{Parent}(x, z)$ says that if x is a grandparent of y, there must be some individual z whose parent is x. First-order logic can also use functions over reals, as seen in satisfiability modulo theory (SMT) [11]. As we illustrated with weighted model integration, which is also defined using SMT, we can express formulas such as $x \leq 5 \land x \geq -5 \land x > y^2$. Here, both the variables are assumed to be nullary functions. But we could also have functions with arguments and nestings of these functions to construct well-defined formulas of the sort: $f(f(x, y), y) \leq y^2$.

We might also be interested in entertaining multiple possible truth assignments to model uncertainty about the environment. For example, there is modal logic [112], which can capture possibilities, beliefs, and intentions [171]. A variant of modal logic with numbers on worlds can lead to probabilistic logics [84], that allow us to reason about probabilities on formulas [67] as well as beliefs about these formulas [66,24].

Beyond these formalisms that map atoms (and by extension, formulas) to binary truth values, there are logics that relax that assumption. Fuzzy logics map Boolean symbols to real numbers, leading to real-valued semantics for non-atomic formulas constructed using connectives. For example, if A and B get values between 0 and 1, then $A \lor B$ gets a value of 1 iff $\max(A, B)$ is 1. Moreover, the conjunction could also get a value between 0 and 1, by way of $\min(A, B)$. Such a definition reduces to the classical semantics when both A and B are assigned 1, in which case the maximum of the two would also be 1.

These are all part and parcel of symbolic logic. The choice of the language, the choice of the semantic rules that we use over the well-defined formulas, along with its computational properties such as decidability are aspects of a logical framework. Moreover, once a logical framework is considered, we could choose to prove logical entailments either by considering assignments to the variables and seeing if the consequent follows, or by applying inference rules established in a proof theory [86]. If we choose to add weights [42], measures [84], or belief functions [60], this then leads to notions such as weighted model counting [6] and algebraic model counting [107], defined over the models of a formula (i.e., possible worlds). Ultimately, we could consider theorem proving [86], model checking [9], SAT solving [11], or model counting [78], depending on the context and application.

Each of these dimensions is already impacting current inquiries into the properties of machine learning models. For example, to tasks from knowledge-based completion to reasoning with ontology triples using neural techniques, there has been development on so-called *neural theorem provers* [137]. These are inspired by Prolog’s proof-theoretic backward chaining mechanism [53] and the aim in those works is to implement that scheme in an end-to-end learning paradigm. Both SAT solving [200] and model counting [73] are important ingredients in state-of-the-art approaches to regularizing neural networks using logical formulas. This is motivated by the need to ensure neural network predictions always satisfy certain domain constraints. Model checking tools are mainstream for checking the robustness of neural networks [81]. There is also some work [195] on studying whether using real-valued fuzzy logics to permit differentiability in neural networks is comparable to differentiability as a result of probabilistic extensions to model counting [73].

In summary, we can explore a variety of logical syntax and semantics, each of which may have interesting interactions with machine learning properties and capabilities.

2.6. Monotoncity

Classical logic is monotonic. That is, if $\alpha_1, \ldots, \alpha_k \models \beta$, then it cannot be the case that adding new knowledge, say, α' forces us to retract β: formally, it has to be that $\alpha_1, \ldots, \alpha_k, \alpha' \models \beta$ also.

John McCarthy was concerned about the problem of monotonicity and wondered how we might deal with exceptions and abnormality. The problem of monotonicity is so ubiquitous, it even comes up in the formulation of automated planning [163]. For example, imagine that you have an action to paint a box blue and another action that pushes the object. Let us say we paint the object and next, we push the object. When we execute the second action, it is implicit that the color of the object does not change. So we would have to somehow codify not only what the effects of the push action are, but also what the non-effects are. And if we start writing down all the non-effects, there could be exponentially many. Moreover, there are various preconditions that must hold for us to be able to push the object. For instance, we should be strong enough to push it, we must not be holding other objects, we are presumably operating under reasonable gravity assumptions, and so on. And if we start expressing all of them, it again looks like a hopeless task. Yet under some assumptions – so-called causal completeness [163] – modelling domains is feasible. These assumptions state that the conditions provided are both necessary and sufficient for describing the action. (These concerns arise in causal modeling in machine learning as well [151], as we need to accurately identify all the parent variables that influence the variable of interest and describe them at the appropriate level of detail.)

If we do not make that assumption, the alternative approach would be to consider a wide range of typical cases, while also accounting for unusual and exceptional cases by incorporating the concept of abnormality. All of this requires notions of non-monotonicity.

It might be interesting to conceptually contrast this to the machine learning approach to dealing with anomalies and exceptions. With learning models, when trained on existing data, they can identify typical patterns and detect abnormalities within that data [108,129]. An outlier is viewed as a data point with atypical features and an unusual label. For instance, while most men in their forties might be categorized as low or middle-income earners, a data point representing a forty-year-old male banker would likely be classified as a high-income earner. Conversely, we might expect a large proportion of high-income earners to be male bankers in their forties, making them the group that deviates from the norm. In this case, they would be the outliers relative to the general population.

Be that as it may, there is no universal mechanism to address default concepts in a general way with such approaches. Moreover, non-monotonic logic reasoning has given us notions like stable model semantics [76] which now powers recent approaches to neuro-symbolic learning [205]. Interestingly, non-monotonic semantics can also allow us to capture cycles in graphs [54], which ordinarily requires recursion using second-order logic [63]. This may be an important aspect as we utilize neural networks for reasoning about large graphs and the web more generally [145]. Thus, attempting to disregard this area of research seems premature.

2.7. Differentiability

Recent approaches to machine learning can be summarized by emphasizing the importance of differentiability as a key concept. However, it is widely held that logic cannot play a role in this. For example, Turing Award winner Yann LeCunn quips [117]:

How can machine reason and plan in ways that are compatible with gradient-based learning?

Our best approaches to learning rely on estimating and using the gradient of a loss, which can only be performed with differentiable architectures and is difficult to reconcile with logic-based symbolic reasoning.

But as indicated by the sections above, this view is simply uniformed. Probabilities as well as real arithmetic can be mapped on to logical expressions and this means that both routes – a probabilistic one [73] and real-valued semantics [195] one – seem to naturally lead to differentiability. Let us elaborate further below.

There has been a historical understanding that logic and probability are compatible with each other [168,159,17]. These include topics such as 0-1 laws for studying the probability of satisfaction of first-order structures [65], the use of probability to compare the fit of logical hypothesis against observations [37], and perhaps most recently, the use of logic-based solvers by means of (weighted) model counting to compute conditional probabilities for Bayesian networks [42].

At this point, there are plenty of approaches that explicitly use logic for the training of neural networks, especially in the context of regularization and differentiability. This started with the work of UCLA's Semantic Loss [73] and KU Leuven's DeepProbLog [128], both of which adjust the loss function of the deep learning model based on a logical encoding of the constraints and program, respectively. This is an end-to-end approach in the sense that the predictions of the neural network are corrected using the logical solver and back-propagated to the network so that the trained network predicts outputs that are compatible with the constraints. There are also recent approaches that are based on real-valued variables, such as in [92] and in [195]. Providing arithmetic constraints to the training of deep learning networks and ensuring consistency with the provided domain knowledge is an important problem for areas like physics [179] and robotics [97].

However, it would be remiss not to point out that just because differentiability seems to be an important ingredient in the training of machine learning models, it does not mean that we expect every scientist in the area of logic to play game. There is still profound and rigorous work to be done on the integration of logical querying (e.g., computational effort needed to evaluate queries on a large knowledge basis [126]) and probability [13], for example. On the representation side, there are important issues to grapple with, such as languages to reason about logic and probability that permit the domain of quantification to be countably infinite (e.g., natural numbers) and uncountable (e.g., reals) sets [124]. Moreover, modal logics like temporal logics and dynamic logics become useful for deep learning-based endeavors as we navigate to more open-ended problems in dynamic domains [118]. For example, in [96], temporal logic formulas are used to train deep reinforcement learning agents. In [176,181], large language models are used to reason about dynamic epistemic properties [21], including the modelling of theory of mind [68]. And in [97], a temporally extended semantic loss function is considered.

An orthogonal direction of work that has recently been considered is the capturing of neural architectures, such as graph neural networks, using fragments of first-order logic [10]. For the purposes of our discussion, it suffices to say that simply focusing on differentiability or differentiable logics does not quite capture the range of questions that one can investigate in the AI landscape. Issues such as expressiveness, computational properties, and the development of hybrid architectures that combine the advantages of logical and uncertain reasoning continue to be valuable areas of research.

It is worth noting that the metalinguistic applications of logic can be both "external" and "internal." In this subsection, we largely discussed the external view that the machine learning system as a whole needs to be understood as a logical theory. This could involve providing a logical semantics with probabilistic programming, or providing a logical language for multiple autonomous learning entities, even logically formalizing machine learning properties such as fairness [20]. However, it is also possible to use logic as a mathematical function inside a machine learning system — that is, applied internally — which is discussed in a few subsequent subsections. In these cases, for example, a logical formula may act as a constraint that could be incorporated into the loss function of learning paradigms, or may serve as an oracle to reason correctly over machine learning predictions. Thus, logic could be used as a mathematical language to understand the system as a whole or as a mathematical function inside a machine learning system.

2.8. What about "human-like" semantic definitions?

The most well-studied semantics, or perhaps more accurately, the most widely-used semantics in computer science, remains classical [34]. That is, atoms are accorded values of either 0 or 1, and so formulas become Boolean functions. If modalities are introduced, such as time and actions [68], then we look at sequences of models: either a linear sequence or a tree-like sequence [163], for example.

But as mentioned above, there are also approaches where a degree of truth is accorded to formulas, either by allowing the atoms themselves to have non-binary values [206] or by according probabilities or other kinds of measures for complex formulas [60].

All of these notions are explored by establishing some kind of well-definedness, and logicians explore the implications of those conditions. For example, intuitionistic logic looks to weaken material implication [61]. Non-classical belief logics control the proof-depth of logical reasoners [125]. Fuzzy logic [206] was initially introduced with the idea that a truth definition needs to be provided to vague notions [70] such as being tall or making water warm.

Be that as it may, there is an informal argument often made that a mathematically rigorous definition of truth is too precise. Perhaps by training neural networks with real-world observations, they might exhibit more human-like reasoning capabilities that eschew a well-defined notion altogether? The evidence for this has not yet been established. Moreover, is such a feature desirable? Let us, for the moment, consider correct reasoning and understand what can be said about deep learning models implicitly inferring logical steps.

2.8.1. Correct reasoning

There are a number of recent papers looking at the reasoning abilities of large language models, which are so-called transformer architectures trained on large troves of textual data [29]. Despite allowing for a number of different ways to backtrack and infer the correct premise for a query (e.g., so-called "chain-of-thought"), as shown in a number of papers, they seem to incorrectly reason in a number of different ways [36,47,189][138, [208]]. For example, sometimes they struggle with symmetry [153,204]. Although newer models are able to recognize an increasing set of patterns and might get logical relationships and connectives, there is little evidence that they are consistently correct — as [106] puts it: "So close, and yet so far!"

Thus, impressive as they are, these models are not reliable [98]. There is a also growing body of recent work on the limitations of formal reasoning with large language models. For example, [182] consider how well LLMs perform with theory of mind reasoning, seen in card games and gossip protocols [68]. In [190], the performance of OpenAI's latest model for reasoning (so-called "o1")[1] for automated planning is considered, and generally poor performance is reported. A study from a team at Apple [138] reports that minor variations to the reasoning questions can lead to dramatic changes in performance, which is problematic. In [208], it is suggested that LLMs learn the statistical properties of logical tests, rather than emulate the correct reasoning function.

In light of these limitations, there is a compelling argument for a neuro-symbolic approach. For instance, implementing a logical error checker as a post hoc mechanism could effectively verify the results, predictions, and completions generated by large language models. For example, a systematic integration of ChatGPT and Wolfram Alpha was recently attempted.[2] More generally, recent approaches seek to incorporate logical solvers as oracles [155] that can validate or disprove the predictions of neural architectures, including large language models [207,149,148,134].

Putting this altogether, the "native" reasoning capabilities of purely neural models seem clearly limited. It is, of course, plausible that a novel training architecture or new types of datasets might provide the right sort of environment for neural models to perform correct reasoning. But for the moment, validating results and/or improving the training of neural architectures using logical solvers—that is, a neuro-symbolic learning pipeline—seems to be the most promising avenue. Kautz makes a stronger claim [106]:

The observation that tools greatly enhance the power of LLMs is not original. Indeed, commercial LLMs already make heavy use of tools – in particular, tools for internet search for the retrieval augmented generation (RAG) paradigm. Kambhampati et al. (2024) recently showed that an LLM can convert planning and verifications problems presented in natural language into formal STRIPS notation and solve them using an external planning system. I go farther than most researchers pursuing the tool approach in

[1] https://openai.com/o1/

[2] https://writings.stephenwolfram.com/2023/03/chatgpt-gets-its-wolfram-superpowers/

that I mean the title of this paper, "Tools Are All You Need", quite literally: a language model augmented with reasoning tools is sufficient to create true artificial intelligence.

He uses "tools" to mean SAT solvers, or other such logical oracles, very much in line with the thrust of this article, and goes on to argue how LLMs "are the only kind of machine learning system that, like humans, can reliably generalize from a *single* example", and how that coupled with logical tools may support general-purpose AI.

2.8.2. The intentional stance

It is worth noting that, strictly speaking, we do not require that the semantics be given by humans, or that they be hand-written. Symbols can obtained from low-level data (via symbol grounding), or from closely related languages [12], or from abstract descriptions [51] of concepts [115]. The use of symbols in AI also does not mean that symbolic logic experts assume humans manipulate symbols in their head. See [121] for philosophical discussions on this point, which can ultimately be tied to the "intentional stance" [55]. The intuition here is that any capability we attribute to an (artificial or human) agent could be understood in terms of intentions, beliefs, and other mental attitudes, which allow us to characterize what the agent is trying to do. It is a pragmatic perspective rather than a literal representation of the agent's behavior model.

There is extensive work on trying to characterize natural language utterances [140], including connectives [87], and their formal counterparts [191]. This also involves the use of terms and formulas, whose meaning may be built up from context and social environment [196]. While the search for a logic that accurately characterizes these kinds of observations with humans is still ongoing, it is worth noting that we do not need a logical knowledge basis to be consistent either. For example, there is work on paraconsistent logics [31].

Ultimately, we have a range of language choices to work with. We may disagree on the semantics, but having a few different systems that can be mathematically studied seems like a good start.

A follow-up question might be to the tune of: does it still make sense to bother with classical semantics? Just as it makes sense to study logic outside the context of differentiability, we would argue the study of classical semantics is also worthwhile in the AI context. Reasons include: (a) it is a well-defined mathematical model, (b) with the use of modalities and/or non-classical semantics, we can relate different systems, (c) we do not really know which semantics best approximates human reasoning, (d) we may not want mathematical truths that play fast and loose with inevitable conclusions just because we think humans might have some cognitive biases and exhibit inconsistent reasoning, and (e) the science of robust AI is still evolving.

3. Logic and learning can be complementary

As already hinted above, symbolic logic can play an important role in training deep learning models but also in integrating reasoning as a post-hoc process or as a metalinguistic paradigm. That is, we can ensure that the distribution of the trained network respects domain constraints [92]. We can extract rules from trained models and reason about them outside the framework of the network [155]. Or we can use the outputs of the network as inputs to a computational paradigm such as probabilistic programming [128]. There is very interesting work on the semantics of programs that inherently support some notion of differentiation [1]. This is an object of intense theoretical study that can have consequences on the types of distributions that are expressible in programming languages [178]. So, this theory has far-reaching effects on what type of probabilistic models can be modelled effectively.

In the second half of the article, we make the following point: symbols and DL need not compete with each other, and can be complementary. Perhaps the most representative example of this is the burgeoning field of neuro-symbolic AI [74], which has come to encompass things like neural program induction [115], neural theorem improvers, and differentiable logics [207]. We consider some other categories below, as usual, with overlap.

3.1. Symbolic logic as meta-theory

An argument made previously [18] is that symbolic logic can be used to formalize notions currently out of the purview of standard machine learning. These include things like the semantics of involved probabilistic programming languages [178] and understanding the limits of differentiable logics [195], but it can also pertain to a range of more exotic topics.

For example, it is very common in AI applications these days to require frameworks for multi-agent reasoning [3]. In explainable AI [83], in particular, we might require that the robot holds beliefs about the human agent [104]. Modal logics study such phenomena. Thus, there has been a significant amount of recent work on incorporating agent modeling into learning frameworks, with multi-agent reinforcement learning being a prominent example [3]. Furthermore, incorporating agent modeling for explainable planning [2] and utilizing user-provided constraints as reward functions in reinforcement learning [96] are topics of study.

Moreover, complex AI systems are not going to be purely based on providing predictions. They will involve search, constraint reasoning, and planning [167]. This has necessitated new approaches for compositionality [178] and modularity [184]. On a related note, it was noted that weighted model counting [78], which provides the foundation for mapping Bayesian inference to SAT solvers, can be upgraded to also reason about maximization and minimization of properties [107], leading to languages where a number of different AI sub-areas, such as search and optimization, can be unified [23].

An orthogonal but very interesting line of research in the recent years looks at the expressiveness of mainstream neural architectures using logical languages. Primarily, they look at fragments of first-order logic to capture (a simplified version of) neural architectures such as transformers [197] and graph neural networks [203]. These investigations have identified that graph neural networks capture fairly limited fragments of first-order logic [10], while attention mechanisms have been shown to be Turing-complete [154]. In the case of graph neural networks, the community is still exploring the implications of these results but it is believed that these architectures may fail in tasks involving queries that require more expressiveness than the fragment they correspond to. So, in this sense, using logical tools to understand neural architectures can have serious implications in terms of how these architectures are being used and in which circumstances they could be considered reliable.

3.2. High-level knowledge

The interplay between reasoning and learning is often compared to Kahneman's [103] famous distinction of *system 1* versus *system 2* type cognition in humans [166]. This is owing to the fact that AI scientists, for a very long time, have been deliberating on the appropriate way to abstract away low-level perception data with high-level concept knowledge, perhaps going back to Shakey [114]. Many "hybrid" formalisms for reasoning with perceptual data attempt to address the interplay between concepts and observations in a systematic way, e.g., [102,144].

Providing mechanisms as well as formal semantics for abstraction remains a topic of theoretical interest even today [14,93]. Roughly, the idea is given a representation R of the high-level model, to find another representation of R' involving low-level data and concepts, such that R and R' agree on atoms under a suitable mapping μ. That is, R entails an atom a iff R' entails $\mu(a)$. In a probabilistic setting, this might mean that we abstract a continuous distribution w.r.t. to evidence in terms of a discrete distribution [94]. There has also been some work on abstracting causal models [15].

In the specific case of deep learning systems, a key agenda point is how to define abstract concepts, whether extracted directly from data or defined externally, in order to coordinate and interoperate with these systems [115,22,35].

More generally, it is widely acknowledged that concepts such as time, abstraction, and causality will play a key role in designing a general-purpose AI [130]. We would expect such an AI to be capable of reasoning with a rich world model, one that can be interpreted by humans [33]. Roughly, the idea is that given some system description, Σ, it is desirable to reason about the following:

1. **Temporal abstractions:** Given two events, e and e', we would like to know which happened earlier, and whether some trigger in e led to whatever happened in e'.
2. **Induction:** Suppose we have a set of events, $p_1, \ldots, p_n$. We would like to find an idealized instance that generalizes these examples, $\tilde{p}$.
3. **Abstraction:** We would like find atomic descriptions, $\hat{p}$, that characterize the interaction between some of those instances (e.g., $p_1 \wedge p_2 \equiv \hat{p}$). The idea, then, is to use the abstract descriptor for increased comprehensibility [94,93].
4. **Causation:** Finally, given a causal chain from events p_1 to p_n in the sense that p_1 or its descendants causes p_n, we would like to understand what would happen if p_i was set to a certain value (*intervention*) or assumed a value not necessarily seen in the data (*counterfactual*).

Although there is some work on providing a causal semantics to deep learning systems [127], it is still in the early years and studied in a limited way. In contrast, we have very well-studied models of time [157] and causality with symbolic calculi [163,85,89]. It seems like a wasted opportunity to not utilise these frameworks simply because they are purely symbolic, and hence deemed "old-fashioned."

As has been the case for many years now, symbols can be used as abstract identifiers for human-in-the-loop systems [104], and/or interactive machine learning especially when you have non-expert stakeholders engaging with predictors trained on high-dimensional data. In particular, there are very concrete examples from the neurosymbolic landscape that particularly highlight the benefits of using symbols. For example, the work on *reward machines* [96] looks to train deep learning-based reinforcement learning agents by means of high-level, temporally extended specifications, such as formulas expressed in linear temporal logic [41]. The propositions of the language are abstract descriptions of properties that can be understood by humans. There is also work on reasoning about neural concepts in a logical language. Although there have been prior works on hybrid formalisms that allow for machine learning constructs to be used in logic [102], recent neurosymbolic approaches such as DeepProbLog [128] allow us to not only include neural concepts as objects in the logical program, but also to reason about this program as signals that could be fed back into the neural network training. This leads to a trained model that provides predictions and learns distributions that are consistent with the logical specification [91].

3.3. Symbolic logic can instantiate new methods of inference

One observation we emphasized earlier is that precisely because of the close relationship between logic and probability [37,192,16], it is possible to use logic-based solvers for doing probabilistic reasoning. This in turn, can mean that logic-based solvers are used in learning modules in probabilistic machine learning [193], or perhaps to reason about the output distributions of neural networks [73].

This is primarily instantiated via weighted model counting [78], which – as discussed above – is an extension of SAT solving to identify all possible satisfying assignments [6]. And as mentioned, there is also an extension of this strategy to deal with continuous properties via so-called weighted model integration [27]. One broader observation here is that because weighted model counting is defined in terms of weights on the possible models of a logical formula, it is possible to use different types of weights. This means a whole range of different computational tasks defined over the models of a logical formula can be approached using the same abstract specification of weighted model counting. This leads to the notion of *algebraic model counting* [107], where instead of sums over the models and products over the weights of literals, we can consider different kinds of corresponding operations such as *maximum* and *minimum* [6].

A notable development in this space is *knowledge compilation* [49]. This stems from the observation that given a probabilistic model, we may have to compute conditional queries repeatedly. Therefore, there have been efforts in representing a logical formula as a data structure that permits the computation of model counting [49], including in the presence of distinct conditional queries, effectively. This development can be coupled with the notion of algebraic model counting [107], but it has also served as a computational backbone for many emerging representations that unify logic and probability, such as relational Bayesian and Markov networks [194] – in addition to classical Bayesian networks [42], of course – and probabilistic logic programming languages like ProbLog [69].

Circuits provide a new way of doing inference with probabilistic models with the following properties: you pay a one-time cost for compiling the representation, such as a Bayesian network, into such a circuit, and then every query afterwards can be done in time polynomial in the size of the circuit. There is also a broader program of learning such circuits directly [122]. The goal is to find an alternative to classical machine learning models with attractive computational properties for inference [198]. This is a new and exciting way of doing probabilistic reasoning and has even led to new approaches to inference in probabilistic programming [95].

3.4. Logical oracles

There is considerable work on verifying neural networks [174] for safety properties [39] as well as robustness [75], where we want to ensure that the prediction of neural networks does not change arbitrarily for small perturbations to the input. Along these lines, there is a new direction of work where logical reasoners serve as oracles to machine learning predictions to ensure that the predictions are consistent.

A representative example here is the contrasting of reasoning capabilities of large-scale learned models, such as large language models, against that of a symbolic oracle. Recent work on Wolfram Alpha [202] looks to integrate an arithmetic solver with the output of ChatGPT so that reasoning outputs are consistent and coherent with mathematical principles. Similarly, although there is some work on how the chain-of-thought prompting approach can lead to better reasoning outputs by large language models, the use of a logical oracle leads to provably correct outputs. The capabilities of ChatGPT, for example, have been directly studied in [71] and [98], and the use of a logical oracle to provide an externally sourced solution to reasoning problems with large language models is considered in [148]. In [181,176], such an approach has been shown to be applicable to involved problems involving the mental states of multiple agents, commonly referred to as the theory of mind [175,68].

Intuitively, the idea here is related in spirit to the investigations on logic-based loss functions [73] because there too, predictions are expected to conform to logical constraints [91].

3.5. Logic benefits from learning

In the article written so far, we have made the case for machine learning benefiting from logical tools and languages. However, on the other hand, looking back to the early days of logical thought, Aristotle argued for the importance of the process of induction [18]. We need mechanisms to learn the general from the particular, which involves generalizing from specific instances to create a generic statement that applies to all instances. That is, a quantified formula that entails all the atoms. In modern AI, this process is a key source of logical knowledge obtained from data [170,52], in addition to information provided by experts [50].

However, if our logical knowledge is to consist of a combination of expert-provided knowledge and knowledge drawn from examples, there are a number of concerns we need to address. For example, how can we ensure that a hypothesis that is consistent with the background knowledge is extracted from the observations [142]? What kind of properties should the resulting knowledge base have [135]? How do we deal with observations that might be incorrect or noisy [7]? How do we ensure that the formula we generalize from the observations captures not only the observations made so far but also the observations we have not yet seen and might encounter in the future [188,101]?

In recent years, a variety of approaches ranging from statistical relational learning [159] to probably-approximate correct (PAC) semantics [158] to neural program induction [115] and neural rule induction [64] have been explored. These approaches utilize state-of-the-art machine learning tools and theory to learn logical expressions. In some cases, noise in the observations is treated by assuming that the observations are drawn from an unknown distribution. In other cases, the generalization capabilities of neural networks are exploited to learn representations that are empirically robust to this noise.

It is now believed that machine learning will likely impact almost all of computer science because it provides a mechanism to construct models from data [173]. This means that we will continue considering combinations of model-based and data-driven domain knowledge in the future. All of this is even more reason to not entertain notions of dichotomy between logic and learning.

4. Concluding thoughts

In this article, we looked at a few of the misunderstandings that arise when considering the relevance and use of symbolic AI in modern AI systems. We hope the reader is convinced that not only does reasoning and learning have significant overlap – including ideas such as model counting appearing in and linking to multiple concerns – but it is also the case that recent advances are exploiting state-of-the-art learning for reasoning (and vice versa), and in the process, improving on the state-of-the-art.

Whether there might be a future architecture that is very close in spirit to current neural models and makes logical tools redundant is yet to be seen. However, as we have argued, it is hard to imagine that, from a theoretical standpoint, logical analysis itself will become redundant, because many of the desired properties sought out are logical in nature. Despite reported advances in the reasoning capabilities of large language models, currently seen as the culmination of large-scale deep learning models, they still struggle with consistency and correctness in both logical and arithmetic problems.

4.1. Other dimensions

We have not discussed a few key issues that are emerging in the AI landscape. With the growing use of AI systems in financial and industrial applications, issues of trustworthiness and responsibility keep coming up [130].

For example, one area where symbolic logic is widely used in many stochastic systems [43] is the verification of safety properties [174], and/or testing for robustness [39]. The idea with safety properties is to ensure that certain regions in a geometric space are avoided because they might represent dangerous operational areas. In the case of robustness, we want to ensure that small perturbations to the input do not dramatically change the prediction from the neural network. It should not come as a surprise that ideas from logic-based computer science, including temporal logic [41] as well as satisfiability modulo theories [11], are the main tools to formalize and investigate these types of properties.

Another interesting avenue for examining trustworthy and responsible AI is understanding the ethical principles and norms under which AI systems should operate [58]. In this subarea, although mainstream models of concepts such as fairness do not necessarily use logic [199], further analysis of how systems could conform to ethical principles is often pursued through symbolic logic [56]. For example, notions such as act-deontology [111] or consequentialism can be formalized as properties that the system's execution should obey [147,201]. There has been work on using symbolic causal models to understand notions of blameworthiness, and the degree of responsibility [45]. Finally, there is considerable recent work on explainable planning [104], where a formal model is used to capture the user's intent and contrast it with the system's understanding of the world in which it operates [175]. For an overview on how knowledge representation can provide much needed frameworks for ethical and trustworthy AI, see [19].

4.2. Neuro-symbolic AI

As we discussed, one area where concerns about the use of logic seem to disappear is neuro-symbolic AI. Neuro-symbolic AI holds a lot of promise because it can offer interesting ways to combine symbolic logic and deep learning, and build on the success of both. And like the maxim: "the whole is greater than the sum of the parts," such an integration may not simply be the communication of outputs in a divorced way, but could involve a deeper type of synthesis [90]. Some approaches have dealt with loss functions, while others have focused on post-hoc logical reasoning or extracting rules from networks. All of these approaches are interesting in their own right.

There is also a trade-off, at least as per our current understanding, between the complexity and level of detail of the logical knowledge and how effectively it can integrate with a learning system. For example, papers focusing on loss functions typically deal with smaller-sized formulas and constraints [92], while works exploring the integration of learning with knowledge graphs often consider ontologies with more than a hundred or even a thousand nodes [145]. Some may argue at this point whether these examples clearly indicate instances of neurosymbolic paradigms exceeding the capabilities of state-of-the-art machine learning. However, this is somewhat of a nebulous measure because state-of-the-art machine learning does encompass various neurosymbolic notions, even if they do not explicitly acknowledge it. Examples range from concept learning [115] to Wolfram Alpha-type integrations with large language models [202].

Of course, with such a diversity of solutions, it may be challenging to determine the correct approach. Perhaps there is no one-size-fits-all solution, and the combination of logic and deep learning can vary depending on the application. Regardless of the specific approach, it is clear that we need to understand the principles of logical languages and semantics to ensure that resulting mathematical objects are well-defined with desired properties. This appreciation is essential for both theoretical exploration and practical applications.

It should be noted that there is a case to be made for expressive representations. For example, some might come away feeling that the best way to approach the future of neuro-symbolic AI is to focus on very limited languages. But such an view may not be fruitful in the long term. For example, it is widely understood that first-order is useful for generalized assertions [119], and modal logics for time and multi-agent beliefs [68]. In general, the language is critical for capturing the domain correctly. In a statement remarkably similar in spirit, Judea Pearl writes [152]:

This is why you will find me emphasizing and reemphasizing notation, language, vocabulary and grammar. For example, I obsess over whether we can express a certain claim in a given language and whether one claim follows from others. My emphasis on language also comes from a deep conviction that language shapes our thoughts. You cannot answer a question that you cannot ask, and cannot ask a question that you have no words for.

4.3. Much to learn

To sum up, there is a lot to be gained by the relating the mathematical foundations of logic and deep learning. And the benefit is not purely for the logician, but also for the deep learning researcher who wants to think more broadly than prediction with big data.

We should, of course, celebrate successes — its neither an accident nor misplaced opportunism that logic/programming language folks are interested in learning and are eager to understand the latest and best [82]. Moreover, what combination of logic and/or learning would be needed for general-purpose AI is not well-understood yet. We cannot point to the exact approach or balance of innateness vs tabula rasa we need for general AI, because we simply do not know. We can only loosely articulate requirements (e.g., correct, fair and safe by design), capabilities (e.g., ability to reason about causality, time and space models) and corresponding desiderata.

Indeed, although AlphaGo and large language models represent a major triumph for AI, these achievements inevitably raise questions about generality and correctness. As mentioned earlier, Kautz [106] argues that a reasoning oracle coupled with a language model might be providing the steps towards general-purpose automated intelligence. Conversely, we may want to be wary of "silver bulletism" —the notion of a single solution addressing all of AI's concerns and capabilities. As Levesque puts it [120]:

As a field, I believe that we tend to suffer from what might be called serial silver bulletism, defined as follows: the tendency to believe in a silver bullet for AI, coupled with the belief that previous beliefs about silver bullets were hopelessly naïve.

Silver bulletism also contributes to the hubris and hype of AI. In view of creating general-purpose, safe, and reliable AI, we need to look at the best of all worlds. And in that regard, the unification of logic and learning continues to bear fruit, of which neuro-symbolic AI is the latest installment.

Acknowledgements

The author is grateful to the Royal Society for funding his University Research Fellowship.

References

[1] M. Abadi and G. D. Plotkin. A simple differentiable programming language. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–28, 2019.

[2] S. V. Albrecht, C. Brewitt, J. Wilhelm, B. Gyevnar, F. Eiras, M. Dobre, and S. Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1043–1049. IEEE, 2021.

[3] S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

[5] R. Azamfirei, S. R. Kudchadkar, and J. Fackler. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2, 2023.

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

[7] 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.

[8] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[9] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[10] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. Reutter, and J.-P. Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[11] C. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability*, chapter 26, pages 825–885. IOS Press, 2009.

[12] S. Bartha, J. Cheney, and V. Belle. One down, 699 to go: or, synthesising compositional desugarings. *Proceedings of the ACM on Programming Languages*, 5(OOPSLA):1–29, 2021.

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

[14] S. Beckers and J. Y. Halpern. Abstracting causal models. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 2678–2685, 2019.

[15] S. Beckers and J. Y. Halpern. Abstracting causal models. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 2678–2685, 2019.

[16] V. Belle. Logic meets probability: Towards explainable ai systems for uncertain worlds. In *IJCAI*, 2017.

[17] V. Belle. Symbolic logic meets machine learning: A brief survey in infinite domains. In *International Conference on Scalable Uncertainty Management*, pages 3–16. Springer, 2020.

[18] V. Belle. Logic meets learning: From aristotle to neural networks. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 78–102. IOS Press, 2021.

[19] V. Belle. Knowledge representation and acquisition for ethical ai: challenges and opportunities. *Ethics and Information Technology*, 25(1):22, 2023.

[20] V. Belle. Toward a logical theory of fairness and bias. *Theory and Practice of Logic Programming*, 23(4):865–883, 2023.

[21] V. Belle, T. Bolander, A. Herzig, and B. Nebel. Epistemic planning: Perspectives on the special issue, 2022.

[22] V. Belle and A. Bueff. Deep inductive logic programming meets reinforcement learning. In *The 39th International Conference on Logic Programming*. Open Publishing Association, 2023.

[23] V. Belle and L. De Raedt. Semiring programming: A semantic framework for generalized sum product problems. *International Journal of Approximate Reasoning*, 126:181–201, 2020.

[24] V. Belle and G. Lakemeyer. Reasoning about probabilities in unbounded first-order dynamical domains. In *IJCAI*, 2017.

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

[26] V. Belle and H. J. Levesque. Reasoning about continuous uncertainty in the situation calculus. In *Proc. IJCAI*, 2013.

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

[28] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[29] A. Birhane, A. Kasirzadeh, D. Leslie, and S. Wachter. Science in the age of large language models. *Nature Reviews Physics*, pages 1–4, 2023.

[30] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint logic programming: syntax and semantics. *TOPLAS*, 23(1):1–29, 2001.

[31] H. A. Blair and V. Subrahmanian. Paraconsistent logic programming. *Theoretical computer science*, 68(2):135–154, 1989.

[32] R. Brachman and H. Levesque. *Knowledge representation and reasoning*. Morgan Kaufmann Pub, 2004.

[33] R. J. Brachman and H. J. Levesque. *Machines like us: toward AI with common sense*. MIT Press, 2022.

[34] A. R. Bradley and Z. Manna. *The calculus of computation: decision procedures with applications to verification*. Springer Science & Business Media, 2007.

[35] A. Bueff and V. Belle. Learning explanatory logical rules in non-linear domains: a neuro-symbolic

approach. *Machine Learning*, pages 1–36, 2024.

[36] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

[37] R. Carnap. *Logical foundations of probability*. Routledge and Kegan Paul London, 1951.

[38] R. Cartuyvels, G. Spinks, and M.-F. Moens. Discrete and continuous representations and processing in deep learning: Looking forward. *AI Open*, 2:143–159, 2021.

[39] M. Casadio, E. Komendantskaya, M. L. Daggitt, W. Kokke, G. Katz, G. Amir, and I. Refaeli. Neural network robustness as a verification property: a principled case study. In *International Conference on Computer Aided Verification*, pages 219–231. Springer, 2022.

[40] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi. Distribution-aware sampling and weighted model counting for sat. *AAAI*, 2014.

[41] K. Chatterjee, M. Chmelik, R. Gupta, and A. Kanodia. Qualitative analysis of pomdps with temporal logic specifications for robotics applications. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 325–330. IEEE, 2015.

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

[43] T. Chen, M. Diciolla, M. Z. Kwiatkowska, and A. Mereacre. A simulink hybrid heart model for quantitative verification of cardiac pacemakers. In *Proceedings of the 16th international conference on Hybrid systems: computation and control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA*, pages 131–136, 2013.

[44] D. Chistikov, R. Dimitrova, and R. Majumdar. Approximate counting in smt and value estimation for probabilistic programs. In *TACAS*, volume 9035, pages 320–334. 2015.

[45] H. Chockler and J. Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[46] T. Crane. The language of thought: No syntax without semantics. 1990.

[47] A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

[48] A. Darwiche. Human-level intelligence or animal-like abilities? *Communications of the ACM*, 61(10):56–67, 2018.

[49] A. Darwiche, P. Marquis, D. Suciu, and S. Szeider. Recent trends in knowledge compilation (dagstuhl seminar 17381). In *Dagstuhl Reports*, volume 7. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[50] E. Davis. *Representations of commonsense knowledge*. Morgan Kaufmann, 2014.

[51] L. De Raedt. Logical settings for concept-learning. *Artificial Intelligence*, 95(1):187–201, 1997.

[52] L. De Raedt, A. Dries, I. Thon, G. Van den Broeck, and M. Verbeke. Inducing probabilistic relational rules from probabilistic examples. In *Proceedings of 24th international joint conference on artificial intelligence (IJCAI)*, volume 2015, pages 1835–1842. IJCAI-INT JOINT CONF ARTIF INTELL, 2015.

[53] L. De Raedt and A. Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100:5–47, 2015.

[54] M. Denecker, M. Bruynooghe, and V. Marek. Logic programming revisited: Logic programs as inductive definitions. *ACM Transactions on Computational Logic*, 2(4):623–654, 2001.

[55] D. C. Dennett. *The intentional stance*. MIT press, 1989.

[56] L. Dennis, M. Fisher, M. Slavkovik, and M. Webster. Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems*, 77:1–14, 2016.

[57] K. R. Dienes. String theory and the path to unification: A review of recent developments. *Physics Reports*, 287(6):447–525, 1997.

[58] V. Dignum. *Responsible artificial intelligence: how to develop and use AI in a responsible way*, volume 1. Springer, 2019.

[59] P. Domingos. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.

[60] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264, 1988.

[61] M. Dummett. The philosophical basis of intuitionistic logic. In *Studies in Logic and the Foundations of Mathematics*, volume 80, pages 5–40. Elsevier, 1975.

[62] K. Ellis, A. Albright, A. Solar-Lezama, J. B. Tenenbaum, and T. J. O'Donnell. Synthesizing theories of human language with bayesian program induction. *Nature communications*, 13(1):5024, 2022.

[63] H. Enderton. *A mathematical introduction to logic*. Academic press New York, 1972.

[64] R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

[65] R. Fagin. Probabilities on finite models. *The Journal of Symbolic Logic*, 41(01):50–58, 1976.

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

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

[68] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

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

[70] K. Fine. Vagueness, truth and logic. 1997.

[71] S. Frieder, L. Pinchetti, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, A. Chevalier, and J. Berner. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*, 2023.

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

[73] K. Gajowniczek, Y. Liang, T. Friedman, T. Zabkowski, and G. Van den Broeck. Semantic and generalized entropy loss functions for semi-supervised deep learning. *Entropy*, 22(3):334, 2020.

[74] A. S. d. Garcez, K. Broda, D. M. Gabbay, et al. *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media, 2002.

[75] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2018.

[76] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP*, pages 1070–1080, 1988.

[77] L. Getoor and B. Taskar. Introduction to statistical relational learning (adaptive computation and machine learning). 2007.

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

[79] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[80] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *Proc. UAI*, pages 220–229, 2008.

[81] T. P. Gros, H. Hermanns, J. Hoffmann, M. Klauck, and M. Steinmetz. Analyzing neural network behavior through deep statistical model checking. *International Journal on Software Tools for Technology Transfer*, 25(3):407–426, 2023.

[82] S. Gulwani. Dimensions in program synthesis. In *PPDP*, pages 13–24. ACM, 2010.

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

[84] J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.

[85] J. Y. Halpern. *Actual causality*. MiT Press, 2016.

[86] J. Y. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *KR*, pages 325–334. Morgan Kaufmann, 1991.

[87] O. T. Heinamaki. *Semantics of English temporal connectives.* The University of Texas at Austin, 1974.

[88] C. S. Herrmann and M. Thielscher. Reasoning about continuous processes. In *AAAI/IAAI, Vol. 1*, pages 639–644, 1996.

[89] C. Hitchcock. Causality: Models, reasoning and inference, 2001.

[90] P. Hitzler. Neuro-symbolic artificial intelligence: The state of the art. 2022.

[91] N. Hoernle, R. M. Karampatsis, V. Belle, and K. Gal. Multiplexnet: Towards fully satisfied logical constraints in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 5700–5709, 2022.

[92] N. Hoernle, R. M. Karampatsis, V. Belle, and K. Gal. Multiplexnet: Towards fully satisfied logical constraints in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5700–5709, 2022.

[93] T. Hofmann and V. Belle. Abstracting noisy robot programs. *AAMAS*, 2023.

[94] 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.

[95] S. Holtzen, G. Van den Broeck, and T. Millstein. Scaling exact inference for discrete probabilistic programs. *Proceedings of the ACM on Programming Languages*, 4(OOPSLA):1–31, 2020.

[96] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

[97] C. Innes and S. Ramamoorthy. Elaborating on learned demonstrations with temporal logic specifications. *arXiv preprint arXiv:2002.00784*, 2020.

[98] M. Jang and T. Lukasiewicz. Consistency analysis of chatgpt. *arXiv preprint arXiv:2303.06273*, 2023.

[99] E. T. Jaynes. How does the brain do plausible reasoning? In *Maximum-entropy and Bayesian methods in science and engineering: Foundations*, pages 1–24. Springer, 1988.

[100] D. Jovanović and L. De Moura. Solving non-linear arithmetic. *ACM Communications in Computer Algebra*, 46(3/4):104–105, 2013.

[101] B. Juba. Implicit learning of common sense for reasoning. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[102] 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.

[103] D. Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[104] S. Kambhampati. Challenges of human-aware ai systems. *AI Magazine*, 41(3), 2020.

[105] M. Katz. Łukasiewicz logic and the foundations of measurement. *Studia logica*, 40:209–225, 1981.

[106] H. Kautz. Tools are all you need. *Proceedings of the 4th Workshop on Logic and Practice of Programming, Held in conjunction with the 40th International Conference on Logic Programming*, 2024.

[107] A. Kimmig, G. V. den Broeck, and L. D. Raedt. Algebraic model counting. *CoRR*, abs/1211.4475, 2012.

[108] V. Kocijan, E. Davis, T. Lukasiewicz, G. Marcus, and L. Morgenstern. The defeat of the winograd schema challenge. *arXiv preprint arXiv:2201.02387*, 2022.

[109] B. Konev, C. Lutz, A. Ozaki, and F. Wolter. Exact learning of lightweight description logic ontologies. *Journal of Machine Learning Research*, 18(201):1–63, 2018.

[110] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[111] B. Krarup, F. Lindner, S. Krivic, and D. Long. Understanding a robot's guiding ethical principles via automatically generated explanations. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 627–632. IEEE, 2022.

[112] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24(1):1–14, 1959.

[113] F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[114] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson. Shakey: from conception to history. *Ai Magazine*, 38(1):88–103, 2017.

[115] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[116] G. Lakemeyer and H. J. Levesque. Cognitive robotics. In *Handbook of Knowledge Representation*, pages 869–886. Elsevier, 2007.

[117] Y. LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62, 2022.

[118] 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.

[119] H. J. Levesque. *Thinking as Computation: A First Course*. MIT Press, 2012.

[120] H. J. Levesque. On our best behaviour. *Artificial Intelligence*, 212:27–35, 2014.

[121] H. J. Levesque and G. Lakemeyer. *The logic of knowledge bases*. The MIT Press, 2001.

[122] Y. Liang, J. Bekker, and G. Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, pages 134–145, 2017.

[123] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[124] 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.

[125] Y. Liu, G. Lakemeyer, and H. J. Levesque. A logic of limited belief for reasoning with disjunctive information. In *KR*, pages 587–597, 2004.

[126] Y. Liu and H. J. Levesque. Tractable reasoning in first-order knowledge bases with disjunctive information. In *Proc. AAAI*, pages 639–644, 2005.

[127] Y. Luo, J. Peng, and J. Ma. When causal inference meets deep learning. *Nature Machine Intelligence*, 2(8):426–427, 2020.

[128] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

[129] G. Marcus. Am i human? *Scientific American*, 316(3):58–63, 2017.

[130] G. Marcus and E. Davis. *Rebooting AI: Building artificial intelligence we can trust*. Vintage, 2019.

[131] J. McCarthy. Programs with common sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, National Physiology Lab, Teddington, England, 1958.

[132] J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28(1):89–116, 1986.

[133] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502, 1969.

[134] A. V. Miceli-Barone, F. Barez, I. Konstas, and S. B. Cohen. The larger they are, the harder they fail: Language models do not recognize identifier swaps in python. *arXiv preprint arXiv:2305.15507*, 2023.

[135] L. Michael. Learning from partial observations. In *IJCAI*, pages 968–974. Citeseer, 2007.

[136] 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.

[137] P. Minervini, M. Bosnjak, T. Rocktäschel, and S. Riedel. Towards neural theorem proving at scale. *arXiv preprint arXiv:1807.08204*, 2018.

[138] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

[139] P. P. Mitra. The circuit architecture of whole brains at the mesoscopic scale. *Neuron*, 83(6):1273–1283, 2014.

[140] R. Moot and C. Retoré. Natural language semantics and computability. *Journal of Logic, Language and Information*, 28:287–307, 2019.

[141] L. Morgenstern and S. A. McIlraith. John McCarthy's legacy. *Artificial Intelligence*, 175(1):1 – 24, 2011.

[142] 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.

[143] N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[144] D. Nitti, V. Belle, T. De Laet, and L. De Raedt. Planning in hybrid relational mdps. *Machine Learning*, 106:1905–1932, 2017.

[145] 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.

[146] Z. Ognjanovic and M. Raškovic. Some first-order probability logics. *Theoretical Computer Science*, 247(1–2):191 – 212, 2000.

[147] M. Pagnucco, D. Rajaratnam, R. Limarga, A. Nayak, and Y. Song. Epistemic reasoning for machine ethics with situation calculus. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 814–821, 2021.

[148] L. Pan, A. Albalak, X. Wang, and W. Y. Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*, 2023.

[149] D. Panas, S. Seth, and V. Belle. Can large language models put 2 and 2 together? probing for entailed arithmetical relationships. *arXiv preprint arXiv:2404.19432*, 2024.

[150] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[151] J. Pearl. *Causality*. Cambridge university press, 2009.

[152] J. Pearl and D. Mackenzie. The book of why.

[153] K. Pei, W. Li, Q. Jin, S. Liu, S. Geng, L. Cavallaro, J. Yang, and S. Jana. Symmetry-preserving program representations for learning code semantics. *arXiv preprint arXiv:2308.03312*, 2023.

[154] J. Pérez, P. Barceló, and J. Marinkovic. Attention is turing-complete. *Journal of Machine Learning*

Research, 22(75):1–35, 2021.

[155] C. Persia and A. Ozaki. Extracting rules from neural networks with partial interpretations. *arXiv preprint arXiv:2204.00360*, 2022.

[156] J. Prado, A. Chadha, and J. R. Booth. The brain network for deductive reasoning: a quantitative meta-analysis of 28 neuroimaging studies. *Journal of cognitive neuroscience*, 23(11):3483–3497, 2011.

[157] A. Prior. *Past, present and future*. Oxford University Press, 1967.

[158] A. Rader, I. G. Mocanu, V. Belle, and B. Juba. Learning implicitly with noisy data in linear arithmetic. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[159] 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.

[160] L. D. Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proc. IJCAI*, pages 2462–2467, 2007.

[161] V. Raman, N. Piterman, and H. Kress-Gazit. Provably correct continuous control for high-level robot behaviors with actions of arbitrary execution durations. In *ICRA*, pages 4075–4081, Karlsruhe, Germany, 2013.

[162] G. N. Reeke and G. M. Edelman. Real brains and artificial intelligence. *Daedalus*, pages 143–173, 1988.

[163] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.

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

[165] J. Rintanen. Planning as satisfiability: Heuristics. *Artif. Intell.*, 193:45–86, 2012.

[166] F. Rossi. Thinking fast and slow in AI: A cognitive architecture to augment both AI and human reasoning (invited talk). In P. Shaw, editor, *30th International Conference on Principles and Practice of Constraint Programming, CP 2024, September 2-6, 2024, Girona, Spain*, volume 307 of *LIPIcs*, pages 2:1–2:1. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[167] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

[168] S. J. Russell. Unifying logic and probability. *Commun. ACM*, 58(7):88–97, 2015.

[169] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

[170] M. Sap, V. Shwartz, A. Bosselut, Y. Choi, and D. Roth. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, 2020.

[171] S. Sardina and Y. Lespérance. Golog speaks the bdi language. In *Programming Multi-Agent Systems*, volume 5919 of *LNCS*, pages 82–99. Springer Berlin Heidelberg, 2010.

[172] T. J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020.

[173] R. B. Shapiro, R. Fiebrink, and P. Norvig. How machine learning impacts the undergraduate computing curriculum. *Communications of the ACM*, 61(11):27–29, 2018.

[174] A. Shih, A. Darwiche, and A. Choi. Verifying binarized neural networks by angluin-style learning. In *Theory and Applications of Satisfiability Testing–SAT 2019: 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9–12, 2019, Proceedings 22*, pages 354–370. Springer, 2019.

[175] M. Shvo, T. Q. Klassen, and S. A. McIlraith. Towards the role of theory of mind in explanation. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems: Second International Workshop, EXTRAAMAS 2020, Auckland, New Zealand, May 9–13, 2020, Revised Selected Papers 2*, pages 75–93. Springer, 2020.

[176] D. Sileo and A. Lernould. Mindgames: Targeting theory of mind in large language models with dynamic epistemic modal logic. *arXiv preprint arXiv:2305.03353*, 2023.

[177] P. Smolensky. Connectionist ai, symbolic ai, and the brain. *Artificial Intelligence Review*, 1(2):95–109, 1987.

[178] S. Staton, H. Yang, F. Wood, C. Heunen, and O. Kammar. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 525–534, 2016.

[179] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[180] L. W. Swanson. *Brain architecture: understanding the basic plan*. Oxford University Press, USA, 2012.

[181] W. Tang and V. Belle. Tom-lm: Delegating theory of mind reasoning to external symbolic executors in large language models. 2024.

[182] W. Tang and V. Belle. Tom-lm: Delegating theory of mind reasoning to external symbolic executors in large language models. In T. R. Besold, A. d'Avila Garcez, E. Jiménez-Ruiz, R. Confalonieri, P. Madhyastha, and B. Wagner, editors, *Neural-Symbolic Learning and Reasoning - 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part II*, volume 14980 of *Lecture Notes in Computer Science*, pages 245–257. Springer, 2024.

[183] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32(4):64–76, 2011.

[184] E. Ternovska and D. G. Mitchell. Declarative programming of search problems with built-in arithmetic. In *Proc. IJCAI*, pages 942–947, 2009.

[185] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[186] J. Tooby, L. Cosmides, and H. C. Barrett. Resolving the debate on innate ideas. *The innate mind: Structure and content*, pages 305–337, 2005.

[187] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[188] L. G. Valiant. Robust logics. In *Proceedings of the thirty-first annual ACM symposium on Theory of Computing*, pages 642–651, 1999.

[189] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*, 2022.

[190] K. Valmeekam, K. Stechly, and S. Kambhampati. Llms still can't plan; can lrms? a preliminary evaluation of openai's o1 on planbench. *arXiv preprint arXiv:2409.13373*, 2024.

[191] J. van Benthem. Semantic parallels in natural language and computation. In *Studies in Logic and the Foundations of Mathematics*, volume 129, pages 331–375. Elsevier, 1989.

[192] J. van Benthem. Against all odds: when logic meets probability. *ModelEd, TestEd, TrustEd: Essays Dedicated to Ed Brinksma on the Occasion of His 60th Birthday*, pages 239–253, 2017.

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

[194] G. Van den Broeck, W. Meert, and J. Davis. Lifted generative parameter learning. In *Statistical Relational Artificial Intelligence, AAAI Workshop*, 2013.

[195] E. van Krieken, E. Acar, and F. van Harmelen. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence*, 302:103602, 2022.

[196] M. van Wijk. Logical connectives in natural language: a cultural evolutionary approach. 2006.

[197] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[198] A. Vergari, Y. Choi, A. Liu, S. Teso, and G. Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. *Advances in Neural Information Processing Systems*, 34:13189–13201, 2021.

[199] S. Verma and J. Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE, 2018.

[200] P.-W. Wang, P. Donti, B. Wilder, and Z. Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pages 6545–6554. PMLR, 2019.

[201] A. Winfield, C. Blum, and W. Liu. Towards an ethical robot: internal models, consequences and ethical action selection. In *Proceedings of the 14th Conference Towards Autonomous Robotic Systems*, pages 85–96, 2014.

[202] S. Wolfram. Wolfram— alpha as the way to bring computational knowledge superpowers to chatgpt. *Stephen Wolfram Writings RSS, Stephen Wolfram, LLC*, 9, 2023.

[203] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[204] S. Yamamoto, K. Kobayashi, and R. Tanaka. An empirical automated evaluation and analysis of symmetrical reasoning in large language models. *Authorea Preprints*, 2024.

[205] Z. Yang, A. Ishay, and J. Lee. Neurasp: Embracing neural networks into answer set programming. In *29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, 2020.

[206] L. A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[207] H. Zhang, J. Huang, Z. Li, M. Naik, and E. Xing. Improved logical reasoning of language models via differentiable symbolic programming. *arXiv preprint arXiv:2305.03742*, 2023.

[208] H. Zhang, L. H. Li, T. Meng, K.-W. Chang, and G. V. d. Broeck. On the paradox of learning to reason from data. *arXiv preprint arXiv:2205.11502*, 2022.