

Knowledge representation and acquisition: Reflections on implicitly learning to reason via PAC-Semantics

Ionela G. Mocanu & Vaishak Belle
University of Edinburgh, UK,

Abstract

Human beings are known for their remarkable ability to comprehend, analyze, and interpret commonsense knowledge. This ability is critical for exhibiting intelligent behavior, often defined as a mapping from beliefs to actions, which has led to attempts to formalize and capture explicit representations in the form of databases, knowledge bases, and ontologies in AI agents.

But in the area of large language models (LLMs), this emphasis might seem unnecessary. After all, these models already capture the extent of human knowledge, and can infer appropriate things from it (presumably) as per some innate logical rules? Perhaps they can be trained to also perform mathematical computations?

Although the consensus on the reliability of such models is still being studied, early results do seem to suggest they do not offer logically and mathematically consistent results. In this short summary article, we articulate the motivations for still caring about logical / symbolic artefacts and representations, and report on recent progress.

1. Introduction

Human beings are known for their remarkable ability to comprehend, analyze, and interpret commonsense knowledge [12]. This ability is critical for exhibiting intelligent behavior, often defined as a mapping from beliefs to actions [13], which has led to attempts to replicate it in AI agents. To create intelligent agents, it is essential to provide them with a way of representing knowledge about the world. In so much as rational behavior is defined by beliefs, intelligent behavior can be argued to depend on an agent's understanding of the world and their capability to form judgments about the different possible states of the world. Not surprisingly, topics such as knowledge acquisition, knowledge representation, and knowledge bases are utmost important.

But in the area of large language models (LLMs), this emphasis might seem unnecessary [9]. After all, these models already capture the extent of human knowledge, and can infer appropriate things from it (presumably) as per some innate logical rules? Perhaps they can be trained to also perform mathematical computations?

In this short summary article, we articulate the motivations for still caring about logical / symbolic artefacts and representations, and report on recent progress.

1.1. Problems at bay: reasoning via learning

Unfortunately, almost all recent results (both anecdotal evaluations on social media) as well rigorous early works seems to indicate that these architectures on their own are not simply not reliable as a source for logically justifiable conclusions.

Preprint submitted to Elsevier

July 6, 2023

For example, the Oxford philosopher Luciano Floridi asks:¹

what's the name of the daughter of Laura's mother?

ChatGPT (even GPT-4, the latest and best comes back with) something to the effect of: "I'm sorry, I don't have personal information about specific individuals." The use of personal pronouns here is misleading [55], but that is a topic for another debate.

Analogously, the computer scientist Vincent Conitzer tried out the below puzzle:²

I put one apple in a bag. Then I put either one pear or two bananas in the bag. Then I remove one piece of fruit. Then I remove all the remaining pieces of fruit from the bag.

Based on what I told you, is it possible to know exactly how many pieces of fruit remain in the bag?

Based on the information provided, it is not possible to know exactly how many pieces of fruit remain in the bag after removing one piece of fruit and then removing all remaining pieces. This is because there are two possible scenarios, but we don't have enough information to determine which one took place:

- 1. One apple and one pear were in the bag. If you removed one piece of fruit, there would be one piece left. After removing all remaining pieces, there would be 0 pieces left.*
- 2. One apple and two bananas were in the bag. If you removed one piece of fruit, there would be two pieces left. After removing all remaining pieces, there would also be 0 pieces left.*

In both scenarios, after removing one piece of fruit and then removing all remaining pieces, there are no pieces of fruit left in the bag.

But apparently in a follow-up reply to that tweet, GPT-4 seems to get it correct.

What can we make of all of this? These models are clearly not reliable, or consistent [29]. Perhaps this what prompted Stephen Wolfram to consider a systematic integration of ChatGPT and Wolfram Alpha.³

There are, of course, plenty of other examples for integrating LLMs and other symbolic solvers, and this is an exciting area for the future of logic, as seen in neuro-symbolic AI [53]. However, the "native" capabilities are clearly limited. An intriguing hypothesis put forward by eminent AI scientist Subbarao Kambhampati is this:⁴

I think many of the claims about LLM's reasoning capabilities miss the point that LLM's are not just trained on "facts" but also, quite often, the deductive closure of those facts. Thus reasoning becomes (approximate) retrieval.

If this is the case, these models do not reason at all, but simply see for patterns of conclusions, which might limit, say, the number of inference steps, or how involved the reasoning is. However, what about consoling ourselves with the idea that the training data might include all such deductions, in which case, LLMs might be sufficient? Sadly, in a critical examination [54], it is shown that LLMs likely pick up unnecessary statistical features of logical inputs, and their logical reasoning abilities may not be sound across different distributions on background theories, and thus, likely not complete.

In what follows, we briefly report on some of our own promising results on learning expressive logical models, but also discuss related work. These results are robust and reliable,

¹<https://twitter.com/Floridi/status/1635951391968567296?s=20>

²<https://twitter.com/conitzer/status/1636156048347111425?s=20>

³<https://writings.stephenwolfram.com/2023/05/the-new-world-of-llm-functions-integrating-llm-technology-into-the-wolfram-language/>

⁴<https://twitter.com/rao2z/status/1666294366720360449?s=20>

but obviously have not enjoyed the scale of LLMs. As a future research agenda, we imagine cross-fertilizations of these areas of research.

2. Going back to symbolic artefacts

One of the key areas of artificial intelligence is concerned with the representation of knowledge and automated reasoning, known as *Knowledge Representation and Reasoning* (KRR). This field focuses on developing systems that express symbolic knowledge, and are capable of reasoning and drawing inferences based on the knowledge they possess, instead of relying solely on explicit programming or heuristics. KRR systems aim to capture the structure of knowledge and the relationships between different pieces of information. This enables the systems to perform complex reasoning tasks and make intelligent decisions [35].

The first term, *knowledge representation*, focuses on the formal representation of the world that captures important properties as well as relationships between objects, events or concepts. Knowledge representation is concerned with formulating a symbolic language that is able to describe the world in a way that is both rich enough to capture important details, but also simple enough for an agent to comprehend and manipulate. This is encoded as a collection of propositions believed by some intelligent agent. We can think of knowledge, first, as a collection of propositions (an abstract entity that can be *true* or *false*, *right* or *wrong*, *factual* or *non-factual*) held by the agent, and secondly, in terms of different ways the world could be, therefore beliefs can be consistent and complete with the real world (so the agent knows everything that is know), inconsistent or just incomplete but correct. This collection of symbols is called the *knowledge base* of the agent. (Ordinarily, we would use propositional or first-order logic, but logical terms for time and actions could also be included [43].)

At the symbol level, there are data structures that capture what the system knows, as well as the reasoning procedures that make what is known available. That is, once knowledge is formally represented, the next step is performing *reasoning* with this information. Reasoning refers to the formal manipulations of the symbols representing the information believed by an agent to produce representations of new knowledge. We imagine a knowledge-based system as being in some sort of abstract epistemic state. It acquires information over time, moves between states (knowing more or less), and uses what it knows to carry out actions and achieve goals. Such an idea can be seen to be in line with the so-called *intentional stance*, as postulated by Daniel Dennett [22]:

Here is how it works: first you decide to treat the object whose behavior is to be predicted as a rational agent; then you figure out what beliefs that agent ought to have, given its place in the world and its purpose. Then you figure out what desires it ought to have, on the same considerations, and finally you predict that this rational agent will act to further its goals in the light of its beliefs. A little practical reasoning from the chosen set of beliefs and desires will in most instances yield a decision about what the agent ought to do; that is what you predict the agent will do.

The reasoning then involves drawing logical conclusions from the available information and using it to make decisions or take future actions. Reasoning is performed through several techniques such as deductive or inductive reasoning.

Deductive reasoning focuses on drawing conclusions from a set of premises using logical rules of inference. For instance, given the premises "*All men are mortal*" and "*Socrates is a man*", applying deductive reasoning leads to the conclusion that "*Socrates is mortal*" Deductive

reasoning is widely used for automatic reasoning systems such as theorem provers or model checkers [27]. They can be used to automatically verify the correctness of a given system or demonstrate its failure by providing a counterexample.

Inductive reasoning starts from specific observations and based on them, draws the hypothesis about future observations. For instance, if a person observes a number of swans, and all swans ever seen by this person are white, this person might conclude that in general, all swans are white. This of course is valid only for the given premises, as it is possible that swans of other color than white exist, but that person has not seen any yet. One of the challenges with this type of reasoning is that it is difficult to determine how much evidence is necessary to support a certain conclusion, i.e. how many swans are enough swans to conclude that all swans are white? In some cases, a small number of observations is sufficient for drawing a reasonable conclusion, while in other cases, a larger amount of evidence is required. Inductive reasoning is widely applied in scientific research, in formulating hypotheses based on observations which are then validated through further experiments. Similarly, in medical diagnosis, inductive reasoning is used for identifying symptoms and patterns in patients' data. And of course, inductive reasoning is also widely applied in machine learning techniques, for developing algorithms that can identify patterns and generalize from input data. For instance, supervised learning involves learning a function that maps an input to its corresponding output from a set of pre-labelled data points. In tasks of natural language processing, inductive techniques are applied to learn a model that can understand the structure and content of a language, such as grammar, or syntax, and then generate natural language. The input data of the input-output format used to generate new hypotheses is also known as training data. The input represents the features or attributes of the data, while the outputs represent the labels or predictions associated with that input.

To contrast, a knowledge base, as defined above, includes information such as facts, concepts, relationships and rules that define the domain of discourse and its properties. Knowledge bases are used for deductive reasoning, where logical inferences are made based on the rules and facts stored in the knowledge bases. In that sense, training data and knowledge bases differ in their contents but also their purposes. The training data is typically specific to a task and is used to produce a model for that task but this is discarded once the model is trained. Whereas the knowledge base is valid across the lifetime of the agent (although it may be updated) and is applied for performing reasoning and action selecting. But all is not well: we will focus next on the knowledge-based systems and the challenges they oppose when being applied to real-world tasks.

2.1. Learning knowledge bases is hard

When dealing with complex domains, the design of a suitable representation language presents inherent challenges. From a semantics standpoint, effective knowledge representation should enable one to represent a broad spectrum of entities and concepts in a precise and unambiguous manner. This formalism should enable us to specify the meaning and consequences of a theory and derive logical consequences from it. In certain cases, a query may yield a response of the form "*I don't know*" indicating a lack of sufficient evidence or information to provide a definitive answer.

Consequently, reasoning under uncertain or incomplete data inputs, like the one just mentioned, requires appropriate handling of such responses. Partial observability arises when some information pertaining to the state of the world is obscured or absent. Reasoning under partial observability presents numerous challenges. Conventional logical reasoning methods, exemplified by theorem provers, often struggle when confronted with extensive knowledge bases and

highly intricate domains. Propositional reasoning is NP-hard, and first-order logical reasoning is undecidable, but Horn logic can be reasoned in polynomial time [11].

To address these limitations, one proposal, but not the only candidate, is the PAC-Semantics framework of Valiant. This framework integrates partially observed examples into the reasoning process, enabling effective reasoning even in scenarios where the available information may be incomplete or invalid. By doing so, it bridges the gap between deductive and inductive reasoning approaches.

Deductive reasoning entails the derivation of logical conclusions from given premises, while inductive reasoning involves generalization from observed examples to make probabilistic inferences. The PAC-Semantics framework capitalizes on the strengths of both deductive and inductive reasoning by incorporating possibly noisy observed data into the reasoning process while maintaining logical soundness and completeness.

2.2. *The approach*

A knowledge-based system contains information expressed using knowledge representation languages, such as first-order logic, description logic, propositional logic, and/or temporal logic. Then, inference rules and reasoning algorithms are used to process information from the knowledge base to arrive at conclusions. Knowledge-based systems are currently used in a wide range of applications, from medical diagnosis and financial analysis to engineering design and various other industries. For example, a knowledge-based system in medical diagnosis might contain information about symptoms or medical history.

In order to effectively deploy knowledge-based systems in the real world, the challenge of knowledge acquisition must be addressed. Knowledge acquisition involves capturing and formalizing knowledge from human experts or from data sources. While knowledge-based systems provide a structured representation of information and relationships, in reality, the content often becomes incomplete or outdated over time. As such, manually updating and adding new knowledge by human experts can be a labour-intensive and error-prone process. As a result, machine learning alternatives have been proposed to deal with the challenge of knowledge acquisition. Training machine learning models on large datasets of patients, for example, will ensure the model is up to date and will help make predictions on new data. However, machine learning has difficulty acquiring rules that feature the kind of exceptions that are prevalent in real-world knowledge. For instance, rules on medical diagnosis are highly context-specific and may have exceptions that arise in certain rare cases [34].

Moreover, it is conjectured to be impossible to reliably learn representations featuring a desirable level of expressiveness. Numerous conventional methods appeal to heuristics without any assurances of robustness. When it comes to decision-making based on complex, domain-specific knowledge, heuristic machine-learning techniques fall short or maybe unreliable.

Many techniques have been developed to learn the logical representation of the hypothesis from input examples, including inductive logic programming (ILP) [38], statistical relational learning (SRL) [40], neurosymbolic AI (NeSy) [28] as well as PAC-semantics based robust learning. There are also related areas such as constraint learning [16].

All these approaches aim to learn logical formulas from examples, but they differ in the underlying assumptions and techniques. Inductive logic programming aims to produce a logical representation of the hypothesis from positive and negative input data, and is based on logical entailment. ILP algorithms operate by searching for the most general set of rules that can explain the positive examples. Constraint learning, on the other hand, aims at learning mathematical constraints from input instances, which are represented as numerical values. ILP is perhaps more

suited for logical reasoning and the acquisition of quantified common-sense statements, whereas constraint learning is widely applied for optimization tasks such as scheduling [2] and resource allocation problems [47]. An instance of a scheduling task is finding a valid shift schedule for the employees while taking into account various constraints such as availability, skill set or preferences. These constraints could take the form of: "the number of hours worked for each employee must not exceed the maximum hours allowed of 16 hours", "each shift must be covered by at least two employees", or "no employee can work more than two consecutive shifts". The first constraint, for instance, is encoded as $\text{maxHours}(E) : \text{employee}(E) \wedge \text{sumHours}(E, H) \wedge (H \leq 16)$, where $\text{employee}(x)$ and $\text{sumHours}(x)$ are predicates that take as arguments the employee variable and the number of hours worked.

PAC-Semantics was introduced by Valiant [52] as a rigorous and general proposal for learning to reason in formal languages. Although weaker than classical entailment, it allows for a powerful model-theoretic framework for answering queries while requiring minimal assumptions about the form of the distribution in question. While PAC-Semantics offers strong guarantees, learning explicit representations is not tractable, even in propositional logic; see discussions in [6]. So it makes sense to assume that one would be capable of achieving more if this requirement to explicitly represent the hypothesis would be mitigated.

One unconventional approach, explored by Khardon and Roth [32] and Juba [30], proposed a solution to such problems by learning to reason directly, bypassing the intractable step of producing an explicit representation of the learned knowledge. Initial works focused on Boolean logics, however, and are therefore limited in terms of expressiveness and applicability. Recent works, surprisingly, on this flavour of "implicit" learning have shown great promise in terms of obtaining polynomial-time results for fragments of first-order logic [6]. The reasoning problem considered is answering queries about formulas based on background knowledge partially represented explicitly as other formulas and partially represented as examples independently drawn from a fixed but unknown probability distribution. This makes the approach very powerful, because most conventional logic-based approaches, such as in Markov logic networks [44] and inductive logic programming [42], including those based on deep learning [23], simply learn representations that fit the data and have nothing to say about unseen data, presumably because there is no underlying model of what the distribution that generates the examples might look like.

It would be interesting, of course, to consider more expressive languages, including ones that reason about mathematics. However, a complex language increases the computational effort, resulting in intractable decision procedures, meaning that their execution time scales exponentially with the size of the input size.

2.3. Continued progress

In very recent work, we examine implicit learning to reason for arithmetic theories [37], including logics considered with satisfiability modulo theory (SMT) solvers [3]. SMT solvers have found extensive application in various domains, such as model checkers [15, 19], verification [36, 17], unit test generators [20], interactive theorem provers for higher-order logic [21, 10], as well as probabilistic inference [14]. Our study demonstrates that for standard fragments of linear arithmetic, learning to reason can be performed efficiently. These results stem from a broader finding: the establishment of an efficient reduction from the logical reasoning problem to a corresponding solver with soundness and completeness properties. In particular, we show that for quantifier-free common fragments, such as difference logic and linear real arithmetic, where sound and complete solvers are known, we obtain implicit learning of constraints in an efficient and robust manner. We show more generally that for languages closed under substitutions

of values for variables, including nonlinear arithmetic, implicit learning can always be added to the sound and complete decision procedures. The learning of logical languages with arithmetic operators and equations has not been discussed before, at least not in a robust manner. In closely related work, but with robustness guarantees, [51] consider the learning of linear expressions over piecewise-continuous distributions, and [5] considers the learning of non-linear arithmetic expressions in a deep inductive logic programming framework.

In later work [41], we extend the implicit learning framework to handle noisy data represented as intervals and threshold uncertainty in the language of linear arithmetic. We establish that this extended language retains its polynomial guarantee despite handling domains of increased structural complexity. Additionally, we present the first empirical investigation of the PACSemantics framework. Using benchmark problems on learning linear program specifications [46], we demonstrate that our implicit approach to learning optimal linear programming objective constraints significantly outperforms the explicit approach of [46] in practice.

Many AI applications nowadays model environments of multiple agents, which act towards achieving goals, either by coordinating or challenging other agents' actions. Modelling the beliefs of agents has also been shown to be very significant for generating explanations [48], and understanding purposeful agents [49, 4]. As AI systems increasingly depend on human input and human interaction [31], it will become important to learn representations that maintain a belief state for the other agents in the environment [50].

In the context of multiagent systems, the development of formal methods for representing and reasoning about interaction among multiple agents in complex systems has emerged as a significant research focus [24], and widely used in diverse areas such as game theory [1], distributed systems [25] and cryptography [26].

Current works on multi-agent logic frameworks mostly are focused on developing formal logics to capture properties of interest. But the problem of learning in such frameworks is at its inception. We extend the PAC-Semantics learning framework for multi-agent epistemic logic.

The formal language employed in this context is propositional logic, augmented with the modal operators to denote knowledge of a specific agent i . For instance, the formula $K_i\phi$ is read as: "the agent i knows (or believes) ϕ to be true."

We consider the problem of answering queries about agents' knowledge, based on an explicitly-encoded knowledge base and (noisy) observations received in real time, which determine the additional knowledge gained by agents. Of course, in line with our discussion above, the "learning" of the background knowledge is only performed implicitly. We consider incomplete knowledge bases, where the information provided to the agent may not determine every fact about the world, and this may involve reasoning about what is believed and also, about what is not believed. We also incorporate the presence of other agents in the environment, whose beliefs may differ arbitrarily from each other and also from the set of facts that are true in the real world.

We show that polynomial-time learnability results can be obtained by means of a modal operator for *only knowing* in addition to the usual operator for knowledge [33]. Only-knowing was originally introduced by Levesque to capture the beliefs of an agent in the sense that its knowledge base is "*all the agent knows*". It was shown that in the multi-agent case [7], the use of only-knowing in the background knowledge can reduce certain types of reasoning tasks into non-modal (e.g., propositional) reasoning. By leveraging that result in our learning approach, it now becomes possible to robust learn to reason (albeit implicitly) with multi-agent epistemic knowledge bases.

2.4. Limitations

Despite the contributions and the promise of the proposed framework, it is important to acknowledge its limitations. For one thing, we have shown that implicit learning could be integrated with a range of languages, from difference logic to epistemic fragments. The framework is currently limited to being able to provide polynomial time results only when the reasoning problem itself can be solved in polynomial time. Perhaps this is not surprising, but it does mean that it may not be well adapted to handle with the intricacies of full fragments. For instance, the framework may face challenges or fail altogether in dealing with second-order logic or temporal logic, which could be critical for some applications. Temporal logic, for instance, deals with reasoning about the ordering and sequencing of events over time. It is crucial in domains that involve temporal dependencies, such as real-time systems, scheduling problems, or temporal databases. Incorporating temporal logic into the framework would require addressing the unique challenges of reasoning about time and causality, which may not be currently supported by the framework.

Producing an explicit representation of the hypothesis provides a clear and interpretable form of knowledge that can be easily examined [23], validated, and understood by human experts [46]. Explicit representation allows for a deeper appreciation of the learned knowledge, its underlying assumptions, and the reasoning processes employed. In many domains, interpretability and explainability are crucial [8], especially when dealing with critical decision-making systems or applications where human trust and accountability are essential [45]. While implicit learning approaches, which focus on learning to reason directly without explicitly representing the knowledge, offer advantages in terms of computational efficiency, they may encounter challenges when it comes to interpretability and transparency.

3. Conclusions

In this paper, we argued for learning explicit representations. We discussed the assumptions behind knowledge representation and why it matters, especially in the context of black box models like large-language models, where there is no explicit representation. It is undeniable that these models would be useful for a number of tasks. But in the context of issues such as explainability, trustworthiness, and having guarantees about the quality of the answers produced, it is still worthwhile to explore explicit representations. Integrating such models with large-language models is also an exciting avenue for research, as seen in [39] and [18], for example.

We briefly covered some of our promising results on PAC-semantics for learning to reason. We also pointed out the limitations of this approach and believe there is hope for robustly learning explicit representations, perhaps under less stringent background assumptions. The PAC-semantics approach, for example, ensures that the learned representation is robust against all possible unknown distributions. Clearly, this is too powerful to learn efficiently and perhaps we need to find compromises that is more pragmatic but nonetheless offers some robustness guarantees for trustworthiness.

References

- [1] R. Aumann and A. Heifetz. Incomplete information. In *Handbook of Game Theory*, vol. 3. Elsevier/North Holland, 2001.
- [2] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer Science & Business Media, 2001.

- [3] C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. *Satisfiability modulo theories*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 825–885. IOS Press, 1 edition, 2009.
- [4] V. Belle, T. Bolander, A. Herzig, and B. Nebel. Epistemic planning: Perspectives on the special issue, 2022.
- [5] V. Belle and A. Bueff. Deep inductive logic programming meets reinforcement learning. In *The 39th International Conference on Logic Programming*. Open Publishing Association, 2023.
- [6] V. Belle and B. Juba. Implicitly learning to reason in first-order logic. *Advances in Neural Information Processing Systems*, 32, 2019.
- [7] V. Belle and G. Lakemeyer. Multiagent only knowing in dynamic systems. *Journal of Artificial Intelligence Research*, 49, 2014.
- [8] V. Belle and I. Papantonis. Principles and practice of explainable machine learning. *Frontiers in big Data*, 2021.
- [9] A. Birhane, A. Kasirzadeh, D. Leslie, and S. Wachter. Science in the age of large language models. *Nature Reviews Physics*, pages 1–4, 2023.
- [10] S. Böhme. *Proving theorems of higher-order logic with SMT solvers*. PhD thesis, Technische Universität München, 2012.
- [11] R. Brachman and H. Levesque. *Knowledge representation and reasoning*. Morgan Kaufmann Pub, 2004.
- [12] R. J. Brachman and H. J. Levesque. Toward a new science of common sense. *arXiv preprint arXiv:2112.12754*, 2021.
- [13] R. J. Brachman, H. J. Levesque, and R. Reiter. *Knowledge representation*. MIT press, 1992.
- [14] R. D. S. Braz, C. O’Reilly, V. Gogate, and R. Dechter. Probabilistic inference modulo theories. *arXiv preprint arXiv:1605.08367*, 2016.
- [15] R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri, and S. Tonetta. The nuxmv symbolic model checker. In *Computer Aided Verification: 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings 26*, pages 334–342. Springer, 2014.
- [16] M.-W. Chang, L.-A. Ratnoff, N. Rizzolo, and D. Roth. Learning and inference with constraints. In *AAAI*, pages 1513–1518, 2008.
- [17] A. Cimatti, S. Mover, and S. Tonetta. Smt-based verification of hybrid systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 2100–2105, 2012.
- [18] R. Confalonieri, T. Weyde, T. R. Besold, and F. M. del Prado Martín. Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artificial Intelligence*, 296:103471, 2021.
- [19] L. Cordeiro, B. Fischer, and J. Marques-Silva. Smt-based bounded model checking for embedded ansi-c software. *IEEE Transactions on Software Engineering*, 38(4):957–974, 2011.
- [20] L. De Moura and N. Bjørner. Z3: An efficient smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems: 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings 14*, pages 337–340. Springer, 2008.
- [21] L. De Moura and G. O. Passmore. The strategy challenge in smt solving. *Automated Reasoning and Mathematics: Essays in Memory of William W. McCune*, pages 15–44, 2013.
- [22] D. C. Dennett. *The intentional stance*. MIT press, 1989.
- [23] R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.
- [24] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
- [25] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *J. ACM*, 37(3):549–587, 1990.
- [26] J. Y. Halpern, R. Pass, and V. Raman. An epistemic characterization of zero knowledge. In *TARK*, pages 156–165, 2009.
- [27] 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.
- [28] P. Hitzler. Neuro-symbolic artificial intelligence: The state of the art. 2022.
- [29] M. Jang and T. Lukasiewicz. Consistency analysis of chatgpt. *arXiv preprint arXiv:2303.06273*, 2023.
- [30] B. Juba. Implicit learning of common sense for reasoning. In *IJCAI*, pages 939–946, 2013.
- [31] S. Kambhampati. Challenges of human-aware ai systems. *AI Magazine*, 41(3), 2020.
- [32] R. Khaldon and D. Roth. Learning to reason. *J. ACM*, 44(5):697–725, 1997.
- [33] H. J. Levesque. All I know: A study in autoepistemic logic. *Artificial Intelligence*, 42(2):263–309, 1990.
- [34] H. J. Levesque. *Common sense, the Turing test, and the quest for real AI*. Mit Press, 2017.
- [35] H. J. Levesque and G. Lakemeyer. *The logic of knowledge bases*. The MIT Press, 2001.
- [36] G. Li and G. Gopalakrishnan. Scalable smt-based verification of gpu kernel functions. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*, pages 187–196, 2010.
- [37] I. G. Mocanu, V. Belle, and B. Juba. Polynomial-time implicit learnability in smt. In *ECAI 2020*, pages 1152–1158.

- IOS Press, 2020.
- [38] 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.
 - [39] C. Persia and A. Ozaki. Extracting rules from neural networks with partial interpretations. *arXiv preprint arXiv:2204.00360*, 2022.
 - [40] D. Poole. Logic, probability and computation: Foundations and issues of statistical relational AI. In *Logic Programming and Nonmonotonic Reasoning*, volume 6645 of *LNCS*, pages 1–9. 2011.
 - [41] A. Rader, I. G. Mocanu, V. Belle, and B. Juba. Learning implicitly with noisy data in linear arithmetic. In *IJCAI*, pages 1410–1417, 2021.
 - [42] L. D. Raedt and K. Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27, 2008.
 - [43] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.
 - [44] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.
 - [45] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
 - [46] E. A. Schede, S. Kolb, and S. Teso. Learning linear programs from data. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1019–1026. IEEE, 2019.
 - [47] A. Schiendorfer. Constraint programming for hierarchical resource allocation. pages 57–68, 01 2014.
 - [48] 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.
 - [49] M. Shvo, T. Q. Klassen, and S. A. McIlraith. Resolving misconceptions about the plans of agents via theory of mind. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 719–729, 2022.
 - [50] D. Sileo and A. Lerno. Mindgames: Targeting theory of mind in large language models with dynamic epistemic modal logic. *arXiv preprint arXiv:2305.03353*, 2023.
 - [51] S. Speichert and V. Belle. Learning probabilistic logic programs over continuous data. In *Inductive Logic Programming: 29th International Conference, ILP 2019, Plovdiv, Bulgaria, September 3–5, 2019, Proceedings 29*, pages 129–144. Springer, 2020.
 - [52] L. G. Valiant. Robust logics. *Artificial Intelligence*, 117(2):231–253, 2000.
 - [53] 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.
 - [54] 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.
 - [55] K. Zhou, D. Jurafsky, and T. Hashimoto. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*, 2023.