Argumentation and Logic Programming for Explainable and Ethical AI* **

Roberta Calegari[1][0000−0003−3794−2942], Andrea Omicini[2][0000−0002−6655−3869], and Giovanni Sartor[1][0000−0003−2210−0398]

[1] Alma AI – Alma Mater Research Institute for Human-Centered Artificial Intelligence , Alma Mater Studiorum—Università di Bologna, Italy
[2] Dipartimento di Informatica – Scienza e Ingegneria (DISI), Alma Mater Studiorum—Università di Bologna, Italy

Abstract. In this paper we sketch a vision of explainability of intelligent systems as a logic approach suitable to be injected into and exploited by the system actors once integrated with sub-symbolic techniques.
In particular, we show how argumentation could be combined with different extensions of logic programming – namely, abduction, inductive logic programming, and probabilistic logic programming – to address the issues of explainable AI as well as some ethical concerns about AI.

Keywords: explainable AI · ethical AI · argumentation · logic programming · abduction · probabilistic LP · inductive LP.

1 Introduction

In the context of the new "AI Era", intelligent systems are increasingly relying on sub-symbolic techniques—such as deep learning (DL) [1, 6]. The opaqueness of most sub-symbolic techniques engenders fears and distrust, thus it has been argued that the behaviour of intelligent systems should be observable, explainable, and accountable—which is the goal of the eXplainable Artificial Intelligence (XAI) field [7, 1].

In this paper we focus on logic-based approaches and discuss their potential in addressing XAI issues especially in pervasive scenarios that can be designed as open multi-agent system (MAS)—the reference for the design of intelligent systems [6, 29, 30].

In particular, this paper proposes a possible architecture for delivering (ubiquitous) symbolic intelligence to achieve explainability in pervasive contexts. Indeed, we believe that the issue of ubiquitous symbolic intelligence is the key to making the environment truly smart and self-explainable. We also think that

* Roberta Calegari and Giovanni Sartor have been supported by the H2020 ERC Project "CompuLaw" (G.A. 833647). Andrea Omicini has been partially supported by the H2020 Project "AI4EU" (G.A. 825619).

the declarativeness and transparency of the approach can lead to the injection of ethical behaviours into the system—according to the studies showing that moral philosophy and psychology for choosing conceptual viewpoints are close to reasoning based on logic programming (LP) [22]. It is worth noting that the proposed vision – and the corresponding architecture – enables on-demand symbolic intelligence injection, only *where* and *when* required. Sub-symbolic techniques – e.g., deep networks algorithms – are therefore part of our vision, and can coexist in the system even in case they are not fully explainable. Consequently, we believe that one of the main requirements of any system is to identify which parts need to be explained – for ethical or legal purposes, responsibility issues, etc. – and which ones can remain opaque.

Logic-based approaches already have a well-understood role in building intelligent (multi-agent) systems; declarative, logic-based approaches have the potential to represent an alternative way of delivering symbolic intelligence, complementary to the one pursued by sub-symbolic approaches [6]. Logic-based technologies address opaqueness issues, and, when suitably integrated with argumentation capabilities, can become the answer to reach features like interpretability, observability, accountability, and explainability. In our vision, explainability can consist of a system's capability of conversing and debating about situations and choices, having the ability able to provide reports and insights into what is happening. An explanation can be seen as a sort of conversation among the person asking for clarification and the system actors – agents, environment and (e-)institution – enabled by the fact that the system is capable of answer and argue about the questions. As far as ethics is concerned, LP has recently been heavily studied by the research community, precisely in relation to the implementation of ethical machines and systems [27].

Argumentation is certainly the spearhead of the proposed approach, but – in order to tackle the requirements of ubiquitous intelligence demanded by AI – it should be strongly intertwined with logic programming and its extensions. In particular, our vision of symbolic intelligence leverages on argumentation, abduction, inductive logic programming, and probabilistic logic programming, along the line of some recent research works—e.g., [15]. Accordingly, in the paper we discuss our vision and how it can build upon some readily available models and technologies, once they are suitably integrated.

2 Logic approaches for XAI

2.1 Why logic?

The main question that has driven us to sharpen this vision is the following: "What is or can be the added value of logic programming for implementing machine ethics and explainable AI?"

The main answer lies back in the three main features of LP: *(i)* being a declarative paradigm, *(ii)* working as a tool for knowledge representation, and *(iii)* allowing for different forms of reasoning and inference. These features lead

to some properties for intelligent systems that have the potential to be critical in the design of ubiquitous intelligence.

Provability. By relying on LP, the models can provide for a well-founded semantics ensuring some fundamental computational properties – such as correctness and completeness. Moreover, extensions can be formalised, well-founded as well, based on recognised theorems—like for instance, correctness of transitive closure, strongly equivalent transformation, modularity and splitting set theorem. Provability is a key feature in the case of trusted and safe systems.

Explainability. The explainability feature is somehow intrinsic in LP techniques. Formal methods for argumentation-, justification-, and counterfactual-based methods are often based on a logic programming approach [14, 23, 27]. These techniques make the system capable to engage in dialogues with other actors to communicate its reasoning, explain its choices, or to coordinate in the pursuit of a common goal. So, the explanation can be a dialogue showing insights on reasoning or, again, the explanation can be the unravel of causal reasoning based on counterfactual. Counterfactuals are the base for hypothetical reasoning, a necessary feature both for explanation and machine ethics. Furthermore, other logical forms of explanation can be envisaged via non-monotonic reasoning and argumentation, through a direct extension of the semantics of LP.

Expressivity and situatedness. As far as the knowledge representation is concerned, the logical paradigm brings non-obvious advantages—beyond the fact of being human-readable. First of all, a logical framework makes it possible to grasp different nuances according to the extensions considered—e.g., nondeterminism, constraints, aggregates [13]. Also, assumptions and exceptions can be made explicit, as well as preferences—e.g., weighted weak constraints [3]. Finally, extensions targeting the Internet of Things can allow knowledge to be situated in order to be able to capture the specificities of the context in which it is located [11]. Expressive, flexible, and situated frameworks are needed to cover various problems and reasoning tasks closely related to each other.

Hybridity. One of the strengths of logic – and of LP specifically – is to make it possible the integration of diversity [10, 28]—e.g., logic programming paradigms, database theory and technologies, knowledge representation, non-monotonic reasoning, constraint programming, mathematical programming, etc. This makes it possible to represent the heterogeneity of the contexts of intelligent systems – also in relation to the application domains – and to customise as needed the symbolic intelligence that is provided while remaining within a well-founded formal framework.

2.2 User requirements for XAI

Before we move into the discussion of the main extensions that a symbolic intelligence engine needs to have in order to inject explainability, let us define what

we should expect from an explainable system and what kind of intelligence the system is supposed to deal with.

- **$\mathbf{R_1}$** First of all, the system should be able to answer *what* questions, i.e., it should provide query answering and activity planning in order to achieve a user-specified goal.
- **$\mathbf{R_2}$** The system should be able to answer *why* questions, i.e., it should provide explanation generation (in the form of text, images, narration, conversation) and diagnostic reasoning.
- **$\mathbf{R_3}$** The system should be able to answer *what if* questions, i.e., it should provide counterfactual reasoning and predictions about what would happen under certain conditions and given certain choices.
- **$\mathbf{R_4}$** The system should be able to answer *which* questions, i.e., it should be able to choose which scenarios to implement, once plausible scenarios have been identified as in the previous point. The choice should result from the system's preferences, which could possibly be user-defined or related to the context.
- **$\mathbf{R_5}$** The system should be able to provide *suggestions*, i.e., to indicate what is better to do given the current state of affair, exploiting hypothetical reasoning.
- **$\mathbf{R_6}$** The system should be able to support two types of intelligence and therefore reasoning, i.e., *reactive reasoning* – related to the data and the current situation – and *deliberative reasoning*—related more to consciousness, knowledge and moral, normative principles.

Even if only **$\mathbf{R_2}$** is strictly and explicitly related to the explainability feature, also the other requirements can help to understand and interpret the system model, so all the above-mentioned requirements can be identified as mandatory for reaching ethical features such as interpretability, explainability, and trustworthiness. According to the requirements, in the following we discuss what logical approach should be part of an engine that enables symbolic intelligence to be injected in contexts demanding for the aforementioned properties.

2.3 Logic approaches and technologies involved for XAI

In our vision, logic programming is the foundation upon which the architecture for a symbolic intelligence engine can be built, enabling an intelligent system to meet the **$\mathbf{R_1}$** requirement. Clearly, enabling different forms of inference and reasoning – e.g., non-monotonic reasoning – paves the way for the possibility to get different answers (appropriate to the context) to the *what* questions. Furthermore, the techniques of inference and reasoning grafted into the symbolic engine make it possible to reason about preferences by meeting requirement **$\mathbf{R_4}$**.

However, LP needs to be extended in order to address explainability in different AI technologies and applications, and to be able to reconcile the two aspects of intelligence present in today's AI systems—namely, *reactive* and *deliberative* reasoning. In particular, in the following we show how argumentation, abduction, induction, and probabilistic LP can be fundamental ingredients to shape explainable and ethical AI.

Argumentation. In this vision, argumentation is the enabler to meet requirement $\mathbf{R_2}$. Argumentation is a required feature of the envisioned symbolic intelligence engine to enable system actors to talk and discuss in order to explain and justify judgements and choices, and reach agreements.

Several existing works set the maturity of argumentation models as a key enabler of our vision [17, 20]. Despite the long history of research in argumentation and the many fundamental results achieved, much effort is still needed to effectively exploit argumentation in our envisioned framework. First, research on formal argumentation has mostly been theoretical: practical applications to real-world scenarios have only recently gained attention, and are not yet reified in a ready-to-use technology [9]. Second, many open issues of existing argumentation frameworks concern their integration with contingency situation and situated reasoning to achieve a blended integration of reactive and deliberative reasoning. Finally, the argumentation architecture should be designed in order to be highly scalable, distributed, open, and dynamic and hybrid approaches should be investigated.

Abduction. Abduction is the enabling technique to meet $\mathbf{R_3}$. Abduction, in fact, allows plausible scenarios to be generated under certain conditions, and enables hypothetical reasoning, including the consideration of counterfactual scenarios about the past. Counterfactual reasoning suggests thoughts about what might have been, what might have happened if any event had been different in the past. What if I have to do it today? What have I learnt from the past? It gives hints about the future by allowing for the comparison of different alternatives inferred from the changes in the past. It supports a justification of why different alternatives would have been worse or not better. After excluding those abducibles that have been ruled out a priori by integrity constraints, the consequences of the considered abducibles have first to be evaluated to determine what solution affords the greater good. Thus, reasoning over preferences becomes possible. Counterfactual reasoning is increasingly used in a variety of AI applications, and especially in XAI [16].

Probabilistic Logic Programming. Probabilistic logic programming (PLP) allows the symbolic reasoning to be enriched with degrees of uncertainty. Uncertainty can be related to facts, events, scenarios, arguments, opinions, and so on. On the one side, PLP allows abduction to take scenario uncertainty measures into account [25]. On the other side, probabilistic argumentation can account for diverse types of uncertainty, in particular uncertainty on the credibility of the premises, uncertainty about which arguments to consider, and uncertainty on the acceptance status of arguments or statements [26].

Reasoning by taking into account probability is one of the key factors that allow a system to fully meet $\mathbf{R_4}$ and $\mathbf{R_5}$, managing to formulate a well-founded reasoning on which scenario to prefer and which suggestions to provide as outcomes.

Inductive LP. Inductive logic programming (ILP) can help us to bridge the gap between the symbolic and the sub-symbolic models—by inserting data and context into the reasoning. As already expressed by $\mathbf{R_6}$, data, context, and reactive reasoning are key features to take into account when designing intelligence. ILP makes it possible learning from data enabling inductive construction of first-order clausal theories from examples and background knowledge. ILP is a good candidate to meet $\mathbf{R_6}$ and preliminary studies show ILP can be the glue between symbolic techniques and sub-symbolic ones such as numerical/statistical machine learning (ML) and deep learning (DL) [2].

All these techniques must be suitably integrated into a unique consistent framework, in order to be used appropriately when needed. They should be involved in the engineering of systems and services for XAI.

2.4 Open challenges

The approaches discussed above are just the starting point for the design and the implementation of the envisioned symbolic engine. Many problems and research challenges remain to be resolved before the architecture can become a ready-to-use framework.

First of all, the model formalisation deserve attention. [15] can provide us with a base for the integration of abduction and PLP, but other approaches integration need to be formalised. Well-founded properties have to been demonstrated as well. Moreover, knowledge extraction and injection techniques have to be explored. [8] depicts a first overview of the main existing techniques but some challenges remain open—in particular, knowledge injection and extraction when dealing with neural networks is a huge problem per se and it is not clear *how* and *where* to inject the symbolic knowledge in such nets [18].

Apart from the extraction of causal cues, there are other cognitive traits that should be taken into account in XAI systems, like, the selection of which kind of explanations. This issue open up different related issues: *(i)* a formal definition of the term explainability, and *(ii)* the selection of a suitable explanation depending on the context. As far as the explainability definition is concerned, the terms "interpretability" and "explainability" are often used carelessly and interchanged in the context of XAI. Although they are closely related and both contributing to the ultimate goal of understandability, it is worth pointing out the differences. On the one side, we borrow the definition of *interpretation* from logic, where the word essentially describes the operation of binding objects to their actual meaning in some context—thus, the goal of interpretability is to convey to humans the meaning hidden into the data [12]. On the other side, we define *explanation* as the act making someone understand the information conveyed in a given discourse—i.e., the goal of explainability is to transfer to the receiver (possibly humans) given information on a semantic level and, for such a reason, we identify explainability as the capability of converse and debate about a situation and on a reached agreement. Not that the distinction between interpretability and explainability shows how most XAI approaches proposed into

the recent literature mostly focus on interpretability. As far as the explanation selection is concerned, cognitive limitations comes into play. For instance, if the explanation is required by a human, due to our cognitive limitations we do not want to be presented with the whole chain of causal connections explaining a given algorithmic decision, but rather users demand for a synthetic explanation going to the core of the causal chain. The individuation of the mechanisms behind such selection is, however, far from trivial and many cognitive techniques should be taken into account [19].

3 System Architecture

Fig. 1 summarises our vision by highlighting the main roles involved in the system as well as the main activity flows. The grey boxes represent the technologies involved in the vision, while arrows represent the expected provided functionalities. The symbolic reasoner embodies the unique framework integrating the aforementioned logic approaches.

On one side, knowledge is collected from various sources – e.g., domain-specific knowledge, ontologies, sensors raw data – and is then exploited by agents that live in a normative environment. Note that we mean to exploit already existing techniques to convert ML knowledge into logic KB [8] and to explore other possibilities – always related to the exploitation of the aforementioned LP approaches – to explain (part of) deep knowledge.

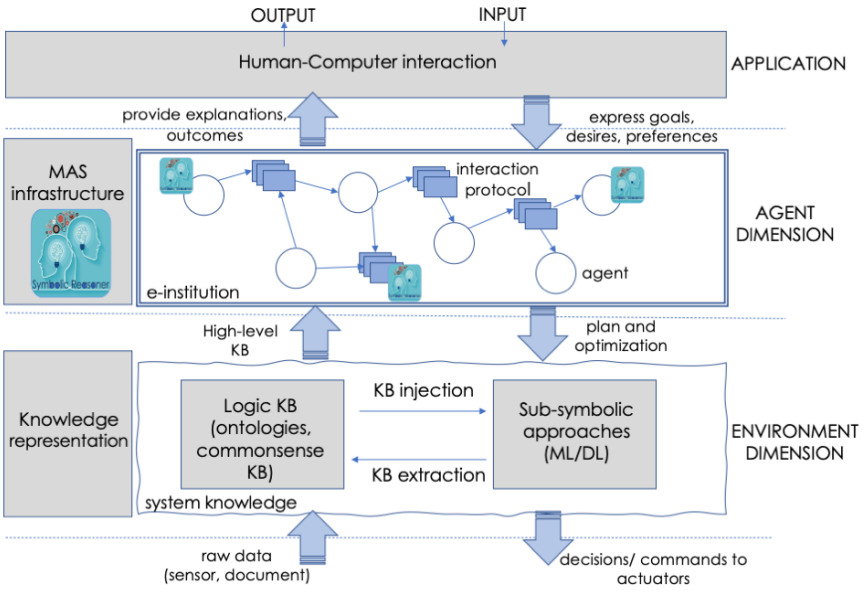

Fig. 1. Main architecture components and techniques for realising the vision.

The cognitive ability of the system is expanded with the concept of symbolic (micro-)intelligence which provides the techniques of symbolic reasoning discussed in Section 2 and tailored to LP. The multi-agent system, also thanks to its rational reasoning and argumentation capabilities, can provide outcomes to the users as well as explanations for their behaviours. On the other side, humans can insert input into the system – like desires, preferences, or goal to achieve – and these are transposed into agents' goal, corresponding activity planning, and lower-level commands for actuators.

The foundation of the vision is to have a symbolic reasoning engine – which carries out the techniques discussed above – to be injectable on-demand into the various system's components—agents and/or environment and/or institutions. Symbolic (micro-)intelligence architecture [4, 21] is exploited to deliver symbolic intelligence according to the new paradigms of AI. The architecture of symbolic (micro-)intelligence should enable – where and when necessary – actions at the micro-level, in order to respond to local and specific needs (for this reason this architecture can be deployed both on cloud and on edge) [5]. Symbolic (micro-)intelligence complements agents' own cognitive processes because it augments the cognitive capabilities of agents, by embodying situated knowledge about the local environment along with the relative inference processes, based on argumentation, abduction, ILP and PLP.

Along this line, our vision stems from two basic premises underpinning the above design: *(i)* knowledge is locally scattered in a distributed environment, hence its situated nature; *(ii)* symbolic capabilities are available over this knowledge, with the goal of extending local knowledge through argumentation, induction, deduction, abduction, and probabilistic reasoning; *(iii)* distributed knowledge can be considered as compartmentalised in distinct knowledge modules and can be used by itself, or by referring to other modules for specific questions (according to the model of modular LP).

4 Preliminary Investigation: Examples

To ground our proposal, let us discuss a preliminary example from a case study in the area of traffic management, considering the near future of self-driving cars. In that scenario, cars are capable of communicating with each other and with the road infrastructure while cities and roads are suitably enriched with sensors and virtual traffic signs able to dynamically interact with cars to provide information and supervision.

Accordingly, self-driving cars need to *(i)* exhibit some degree of intelligence for taking autonomous decisions; they need to *(ii)* converse with the context that surrounds them, *(iii)* have humans in the loop, *(iv)* respond to the legal setting characterising the environment and the society, and *(v)* offer explanations when required—e.g., in case of accidents to determine causes and responsibilities. Fig. 2 (left) contains a possible example of the logical knowledge that, despite its simplicity, highlights the main different sources of knowledge taken into account in such a scenario. First of all, knowledge includes data collected by

vehicle sensors as well as the beliefs of vhicles—possibly related to the outcome of a joint discussion among other entities in the system. Then, commonsense rules enrich the system knowledge, for instance, linking perceptions to beliefs about the factual situations at stake. Also, commonsense rules can state general superiority relations, such as that sensors' perceptions must be considered prevailing over vehicles' beliefs. An additional source of knowledge is e-institution knowledge. Loosely speaking, e-institutions are computational realisations of traditional institutions that incarnate the global system norms as global, national, state, and local laws, regulations, and policies. For instance, the e-institution knowledge defined in Fig. 2 declares that general speed limit – according to Germany federal government – is 100 km/h outside built-up areas (no highways). In addition, a general norm is stated by the e-institution declaring that the overtake is permitted only if it is not raining. Another possible source of knowledge is situated knowledge collected by the surrounding context (infrastructure) that can include specific local rules stating exceptions to the general e-institutions rules. For instance, in the example, situated knowledge states that in the road being represented the general speed limit only applies if it does not rain, otherwise vehicles must slow down to 60 km/h. Note that in the example we list all the different kinds of knowledge in a unique file, but a suitable technology that embodies the envisioned architecture needs to manage different modules and to combine them—depending on the situation.

Fig. 2 (right) shows some system outcomes, depending on the situation. All examples have been implemented and tested on the preliminary implementation of the system—namely, Arg-tuProlog [24].[3] Arg-tuProlog – designed according to the vision discussed in this paper – is a lightweight modular argumentation tool that fruitfully combines modular logic programming and legal reasoning according to an argumentation labelling semantics in which any statement (and argument) is associated with one label which is IN, OUT, UND, respectively meaning that the argument is accepted, rejected, or undecided. Example 1 is run without taking into account the superiority relation of perceptions over beliefs. In this situation, beliefs and perceptions are in conflict and no decision can be taken by the system, i.e., vehicles can base their decision only by taking into account the e-institution obligation and cannot be sure on the permission of overtaking. Example 2, instead, takes superiority relation into account, and according to the fact that sensor perception imposes a speed limit of 60 km/h and negate permission to overtake. The argumentation process among the system actors makes them meet on the conclusion that it rains, so both vehicles, despite their beliefs, will set the maximum speed to 60 km/h. Conversely, Example 3 is run by negating rain perception. The system then recognises that it is not raining, so vehicle speed can be set to 100 km/h, and overtakes are allowed.

The examples discussed are just a simplification of the scenario but already illustrate the potential of rooting explanation in LP and argumentation. A first explanation is provided by the argumentation labelling which allows correlating arguments (and statements) accepted as plausible to a graph of attacks,

[3] https://pika-lab.gitlab.io/argumentation/Arg-tuProlog/

```
%******** SYSTEM KB ********************
%******** ************ ****************

%** PERCEPTIONS and BELIEFS **
pr1:[] => perception(rain).

b1: [] => belief(agent1, rain).
b2: [] => -belief(agent2, rain).


%** GENERAL-COMMONSENSE KB **
% perceptions/beliefs translation
r1: perception(X) => fact(X).
r2: -perception(X) => -fact(X).

r3: belief(A, X) => fact(X).
r4: -belief(A, X) => -fact(X).

%** GENERAL-COMMONSENSE KB **
% perceptions are superior to beliefs
sup(r1,r3).
sup(r1,r4).
sup(r2,r3).
sup(r2,r4).


%** e-INSTITUTION RULES **
% permissions and obligations
o1: [] => o(max_speed(100)).
p1: -fact(rain) => p(overtaking).

%** SITUATED LOCAL KB **
% specific road obligation
% if rains max speed 60 km/h
r5: fact(rain) => speed(60).
r6:-fact(rain),o(max_speed(X))=>speed(X).
```

```
%**** Example 1 ****
IN(accepted) =======>
[obl, [max_speed(100)]]
[neg, belief(agent2, rain)]
[belief(agent1, rain)]
[perception(rain)]
UND(undecided) ====>
[fact(rain)] [fact(rain)]
[neg, fact(rain)][speed(60)]
[speed(100)] [speed(60)]
[perm, [overtaking]]

%**** Example 2 ****
IN(accepted) =======>
[speed(60)][speed(60)]
[obl, [max_speed(100)]]
[fact(rain)][fact(rain)]
[neg, belief(agent2, rain)]
[belief(agent1, rain)]
[perception(rain)]
OUT (rejected) ======>
[speed(100)][neg, fact(rain)]
[perm, [overtaking]]

%**** Example 3 ****
IN(accepted) =======>
[speed(100)][speed(100)]
[perm, [overtaking]]
[obl, [max_speed(100)]]
[neg, fact(rain)][neg, fact(rain)]
[neg, belief(agent2, rain)]
[belief(agent1, rain)]
[neg, perception(rain)]
[perm, [overtaking]]
OUT (rejected) ======>
[speed(60)][fact(rain)]
```

Fig. 2. Example of system knowledge in the self-driving cars scenario, implemented in Arg2P (left). Arg2P system outcomes in three discussed examples (right).

superiority and non-defeasible rules, detailing the system reasoning. If we think about how the scenario could be enriched through abducible and counterfactual enabling a what-if analysis of different scenarios, the possibilities of the system to be explainable become manifold. Furthermore, probabilistic concepts make it possible to stick weight on assumptions, rules and arguments, for instance, agents' beliefs can be weighted according to the social credibility of each of them—possibly measured on numbers of sanctions or whatever. Ethics behaviours can be computed as well – in a human-readable way – preferring, for instance, to minimize the number of deaths in case of accidents. Interesting discussions on the moral choices of the system can be introduced and compared exploiting what-if analysis.

5 Conclusion

The paper presents a vision on how explainability and ethical behaviours in AI systems can be linked to logical concepts that find their roots in logic programming, argumentation, abduction, probabilistic LP, and inductive LP. The proposed solution is based on a (micro-)engine for injecting symbolic intelligence where and when needed. A simple example is discussed in the scenario of the self-driving car, along with its reification on a (yet preliminary) technology—namely Arg-tuProlog. However, the discussion and the corresponding example already highlight the potential benefits of the approach, once it is fruitfully integrated with the sub-symbolic models and techniques exploited in the AI field. In particular, the analysis carried out in the paper points out the key requirements of explainable and ethical autonomous behaviour and related them with specific logic approaches. The results presented here represent just a preliminary exploration of the intersection between LP and explainability, but it have the potential to work as a starting point for further research.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
2. Belle, V.: Symbolic logic meets machine learning: A brief survey in infinite domains. In: Davis, J., Tabia, K. (eds.) International Conference on Scalable Uncertainty Management. Lecture Notes in Computer Science, vol. 12322, pp. 3–16. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58449-8_1
3. Borning, A., Maher, M.J., Martindale, A., Wilson, M.: Constraint hierarchies and logic programming. In: Levi, G., Martelli, M. (eds.) Sixth International Conference on Logic Programming. vol. 89, pp. 149–164. MIT Press, Lisbon, Portugal (Jun 1989)
4. Calegari, R.: Micro-Intelligence for the IoT: Logic-based Models and Technologies. Ph.D. thesis, Alma Mater Studiorum—Università di Bologna, Bologna, Italy (2018). https://doi.org/10.6092/unibo/amsdottorato/8521

5. Calegari, R., Ciatto, G., Denti, E., Omicini, A.: Engineering micro-intelligence at the edge of CPCS: Design guidelines. In: Internet and Distributed Computing Systems (IDCS 2019), Lecture Notes in Computer Science, vol. 11874, pp. 260–270. Springer (10–12 Oct 2019). https://doi.org/10.1007/978-3-030-34914-1_25
6. Calegari, R., Ciatto, G., Denti, E., Omicini, A.: Logic-based technologies for intelligent systems: State of the art and perspectives. Information **11**(3), 1–29 (Mar 2020). https://doi.org/10.3390/info11030167, Special Issue "10th Anniversary of Information—Emerging Research Challenges"
7. Calegari, R., Ciatto, G., Mascardi, V., Omicini, A.: Logic-based technologies for multi-agent systems: A systematic literature review. Autonomous Agents and Multi-Agent Systems **35**(1), 1:1–1:67 (2021). https://doi.org/10.1007/s10458-020-09478-3, collection "Current Trends in Research on Software Agents and Agent-Based Software Development"
8. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for XAI: A survey. Intelligenza Artificiale **14**(1), 7–32 (2020). https://doi.org/10.3233/IA-190036
9. Calegari, R., Contissa, G., Lagioia, F., Omicini, A., Sartor, G.: Defeasible systems in legal reasoning: A comparative assessment. In: Araszkiewicz, M., Rodríguez-Doncel, V. (eds.) Legal Knowledge and Information Systems. JURIX 2019: The Thirty-second Annual Conference, Frontiers in Artificial Intelligence and Applications, vol. 322, pp. 169–174. IOS Press (11-13 Dec 2019). https://doi.org/10.3233/FAIA190320
10. Calegari, R., Denti, E., Dovier, A., Omicini, A.: Extending logic programming with labelled variables: Model and semantics. Fundamenta Informaticae **161**(1-2), 53–74 (Jul 2018). https://doi.org/10.3233/FI-2018-1695, Special Issue on the 31th Italian Conference on Computational Logic: CILC 2016
11. Calegari, R., Denti, E., Mariani, S., Omicini, A.: Logic programming as a service. Theory and Practice of Logic Programming **18**(3-4), 1–28 (2018). https://doi.org/10.1017/S1471068418000364, Special Issue "Past and Present (and Future) of Parallel and Distributed Computation in (Constraint) Logic Programming"
12. Ciatto, G., Calegari, R., Omicini, A., Calvaresi, D.: Towards XMAS: eXplainability through Multi-Agent Systems. In: Savaglio, C., Fortino, G., Ciatto, G., Omicini, A. (eds.) AI&IoT 2019 – Artificial Intelligence and Internet of Things 2019, CEUR Workshop Proceedings, vol. 2502, pp. 40–53. Sun SITE Central Europe, RWTH Aachen University (Nov 2019), http://ceur-ws.org/Vol-2502/paper3.pdf
13. Dyckhoff, R., Herre, H., Schroeder-Heister, P. (eds.): Extensions of Logic Programming, 5th International Workshop, ELP'96, Lecture Notes in Computer Science, vol. 1050. Springer, Leipzig, Germany (Mar 1996). https://doi.org/10.1007/3-540-60983-0
14. Esposito, F., Fanizzi, N., Iannone, L., Palmisano, I., Semeraro, G.: A counterfactual-based learning algorithm for $\mathcal{ALC}$ description logic. In: Bandini, S., Manzoni, S. (eds.) Advances in Artificial Intelligence. AI*IA 2005. Lecture Notes in Computer Science, vol. 3673, pp. 406–417. Springer Berlin Heidelberg (2005). https://doi.org/10.1007/11558590_41
15. Ferilli, S.: Extending expressivity and flexibility of abductive logic programming. Journal of Intelligent Information Systems **51**(3), 647–672 (2018). https://doi.org/10.1007/s10844-018-0531-6
16. Fernández, R.R., de Diego, I.M., Aceña, V., Fernández-Isabel, A., Moguerza, J.M.: Random forest explainability using counterfactual sets. Information Fusion **63**, 196–207 (2020). https://doi.org/10.1016/j.inffus.2020.07.001

17. Hulstijn, J., van der Torre, L.W.: Combining goal generation and planning in an argumentation framework. In: Hunter, A. (ed.) International Workshop on Nonmonotonic Reasoning (NMR'04). pp. 212–218. Pacific Institute, Whistler, Canada (Jan 2004)
18. Kemker, R., McClure, M., Abitino, A., Hayes, T., Kanan, C.: Measuring catastrophic forgetting in neural networks. In: McIlraith, S.A., Weinberger, K.Q. (eds.) AAAI Conference on Artificial Intelligence. pp. 3390–3398. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16410
19. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019). https://doi.org/10.1016/j.artint.2018.07.007
20. Modgil, S., Caminada, M.: Proof theories and algorithms for abstract argumentation frameworks. In: Simari, G., Rahwan, I. (eds.) Argumentation in artificial intelligence, pp. 105–129. Springer, Boston, MA (2009). https://doi.org/10.1007/978-0-387-98197-0_6
21. Omicini, A., Calegari, R.: Injecting (micro)intelligence in the IoT: Logic-based approaches for (M)MAS. In: Lin, D., Ishida, T., Zambonelli, F., Noda, I. (eds.) Massively Multi-Agent Systems II, Lecture Notes in Computer Science, vol. 11422, chap. 2, pp. 21–35. Springer (May 2019). https://doi.org/10.1007/978-3-030-20937-7_2, International Workshop, MMAS 2018, Stockholm, Sweden, July 14, 2018, Revised Selected Papers
22. Pereira, L.M., Saptawijaya, A.: Programming Machine Ethics, Studies in Applied Philosophy, Epistemology and Rational Ethics (SAPERE), vol. 26. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29354-7
23. Pereira, L.M., Saptawijaya, A.: Counterfactuals, logic programming and agent morality. In: Urbaniak, R., Payette, G. (eds.) Applications of Formal Philosophy, Logic, Argumentation & Reasoning (LARI), vol. 14, pp. 25–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58507-9_3
24. Pisano, G., Calegari, R., Omicini, A., Sartor, G.: Arg-tuProlog: A tuProlog-based argumentation framework. In: Calimeri, F., Perri, S., Zumpano, E. (eds.) CILC 2020 – Italian Conference on Computational Logic. Proceedings of the 35th Italian Conference on Computational Logic. CEUR Workshop Proceedings, vol. 2719, pp. 51–66. Sun SITE Central Europe, RWTH Aachen University, CEUR-WS, Aachen, Germany (13-15 Oct 2020), http://ceur-ws.org/Vol-2710/paper4.pdf
25. Poole, D.: Logic programming, abduction and probability. New Generation Computing **11**(3–4), 377 (1993). https://doi.org/10.1007/BF03037184
26. Riveret, R., Oren, N., Sartor, G.: A probabilistic deontic argumentation framework. International Journal of Approximate Reasoning **126**, 249–271 (2020). https://doi.org/https://doi.org/10.1016/j.ijar.2020.08.012
27. Saptawijaya, A., Pereira, L.M.: From logic programming to machine ethics. In: Bendel, O. (ed.) Handbuch Maschinenethik, pp. 209–227. Springer VS, Wiesbaden (2019). https://doi.org/10.1007/978-3-658-17483-5_14
28. Vranes, S., Stanojevic, M.: Integrating multiple paradigms within the blackboard framework. IEEE Transactions on Software Engineering **21**(3), 244–262 (1995). https://doi.org/10.1109/32.372151
29. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: theory and practice. The Knowledge Engineering Review **10**(2), 115–152 (1995). https://doi.org/10.1017/S0269888900008122
30. Xhafa, F., Patnaik, S., Tavana, M. (eds.): Advances in Intelligent Systems and Interactive Applications: Proceedings of the 4th International Conference on In-

telligent, Interactive Systems and Applications (IISA2019), Advances in Intelligent Systems and Computing, vol. 1084. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-34387-3