

Article

# Logic-Based Technologies for Intelligent Systems: State of the Art and Perspectives

Roberta Calegari <sup>1,\*</sup> , Giovanni Ciatto <sup>2</sup> , Enrico Denti <sup>3</sup>  and Andrea Omicini <sup>2</sup> 

<sup>1</sup> Alma AI—Alma Mater Research Institute for Human-Centered Artificial Intelligence, Alma Mater Studiorum—Università di Bologna, 40121 Bologna, Italy

<sup>2</sup> Dipartimento di Informatica—Scienza e Ingegneria (DISI), Alma Mater Studiorum—Università di Bologna, 47522 Cesena, Italy; giovanni.ciatto@unibo.it (G.C.); andrea.omicini@unibo.it (A.O.)

<sup>3</sup> Dipartimento di Informatica—Scienza e Ingegneria (DISI), Alma Mater Studiorum—Università di Bologna, 40136 Bologna, Italy; enrico.denti@unibo.it

\* Correspondence: roberta.calegari@unibo.it

Received: 25 February 2020; Accepted: 18 March 2020; Published: 22 March 2020



**Abstract:** Together with the disruptive development of modern sub-symbolic approaches to artificial intelligence (AI), symbolic approaches to classical AI are re-gaining momentum, as more and more researchers exploit their potential to make AI more comprehensible, explainable, and therefore trustworthy. Since logic-based approaches lay at the core of symbolic AI, summarizing their state of the art is of paramount importance now more than ever, in order to identify trends, benefits, key features, gaps, and limitations of the techniques proposed so far, as well as to identify promising research perspectives. Along this line, this paper provides an overview of logic-based approaches and technologies by sketching their evolution and pointing out their main application areas. Future perspectives for exploitation of logic-based technologies are discussed as well, in order to identify those research fields that deserve more attention, considering the areas that already exploit logic-based approaches as well as those that are more likely to adopt logic-based approaches in the future.

**Keywords:** symbolic AI; logic-based technologies; intelligent systems

## 1. Introduction

Artificial intelligence (AI) is getting evergrowing attention from both academia and industry, in terms of resources, economic impact, available technologies and widespread adoption in virtually any application area. In fact, more and more industries are adopting and applying state-of-the-art AI techniques, to actively pursue challenging business objectives.

Such a general interest, and technology adoption, has been favored by two main ingredients: (i) the development of advanced technologies even at the micro scale, and (ii) the availability of large amounts of data in the environment around us to learn from. These ingredients have boosted *sub-symbolic* AI techniques, such as machine learning (ML)—there including deep learning and neural networks—aimed at exploiting big data to make predictions and take autonomous decisions—in contrast to the more long-established *symbolic* techniques, based on the formal representation of knowledge and its elaboration via explicit reasoning rules.

The increasing role of intelligent systems in human society, however, raises unprecedented issues about the need to *explain* the behavior, or the result of, intelligent systems—in the sense of being capable of *motivating* their decision and make the underlying decision process *understandable* by human beings: this is where sub-symbolic techniques, despite their efficiency, fall short. This is especially relevant when AI is exploited in the context of human organizations meant at providing public services—such as, e.g., health care/diagnostic systems or legal advice. There is therefore an emerging need to reconcile

and synthesize symbolic and sub-symbolic techniques, exploiting the first to *explain* the latter—the scope of *eXplainable Artificial Intelligence* (XAI) [1].

Therefore, just when AI’s general focus is on sub-symbolic techniques, symbolic approaches are re-emerging as the means to bring AI closer to the human understanding, helping humans to overcome fears and ethical issues by providing explainability, observability, interpretability, responsibility, and trustability. In this context, *logic-based* approaches, despite their age, are finding a new youth, for several reasons—first of all, because of their closer relation to the human cognitive process. Moreover, their role in (not just computer) science is well understood—think for instance of the formal study of programs and semantics in computational models, computational logic, inference as computation, logic programming, and automatic theorem proving [2,3]. Last but not least, logic-based approaches have long been at the center of many successful agent-based models and technologies: indeed, agents reason through logic, and plan and coordinate through logical processes [4–6]. Overall, from the XAI spark, wider perspectives are raising also beyond the symbolic/sub-symbolic dichotomy, yet all sharing a logic-based root.

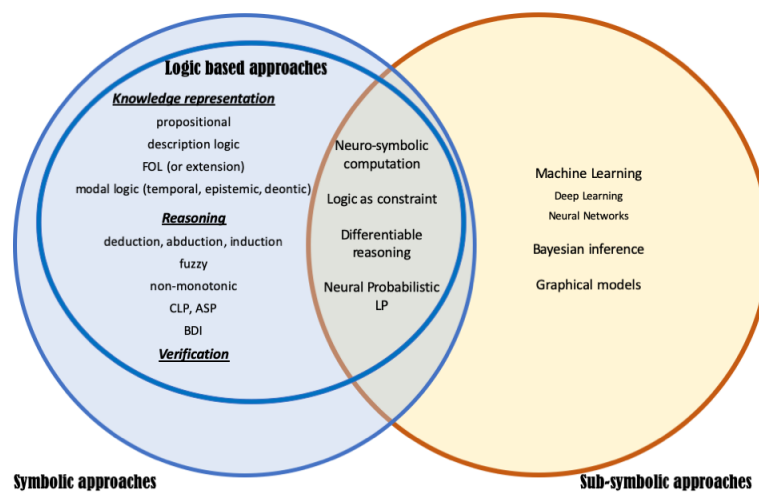
Accordingly, in this paper we focus on the role that logic technologies have played over the years and are going to play in the forthcoming AI landscape—in particular, for the engineering of intelligent systems. We leverage on that analysis to identify the most promising research perspectives and the open issues characterizing the current state of the art.

## 2. State of the Art: 60 Years in a Nutshell

According to Encyclopædia Britannica (<https://www.britannica.com/topic/logic>), *logic* can be defined as “the study of correct *reasoning*, especially as it involves the drawing of *inferences*”. There, *reasoning* refers to one of the most complex and fascinating capability of the human mind, whereas *inference* is one particular way of reasoning—indeed, a *principled* way of reasoning, which consists of drawing *conclusions* out of *premises* following a set of *rules*. Hence its relevance in science, where it can be seen as the main tool to understand the world and gain new knowledge about it, *rigorously*.

In the next sections we quickly overview the main logic-based formalism, technologies, and research lines. In particular, from the technological viewpoint, the logic-based technologies developed in over 60 years can be classified as technologies capable of responding to the needs of (i) knowledge representation, (ii) reasoning, (iii) model-checking and verification—the first two leading in their turn to *expert systems* and *intelligent systems*, which aim to mimic the human behavior.

Figure 1 (left) summarizes this classification, discussed in detail in the following.



**Figure 1.** A graphical view of symbolic and sub-symbolic approaches highlighting the role of logic-based technologies along with their main classification.

## 2.1. The Premise: Computational Logic

In computer science (CS) and artificial intelligence (AI), logic is perhaps the main means for endowing machines with human-like reasoning capabilities, as it is rigorous enough to express *computations*, and represent knowledge in a human-understandable way. In decades, the contribution of logic in CS and AI has widened enough to be considered a discipline by itself—*computational logic* (CL). Logic in CL has then been exploited for diverse purposes—for instance, as a means to (i) compute, (ii) represent computation, (iii) reason about computational systems.

Several logic formalisms exist in CL. They all leverage on some sort of *inference rule* aimed at drawing new *consequences* from a corpus of *premises*. Both premises and consequences represent chunks of useful knowledge expressed as valid *formulas* according to some language of choice. Therefore, the language adopted to represent them, as well as the rules exploited to draw inferences, are of paramount importance when it comes to define logic formalisms. This is why formalisms in CL essentially differ (i) in the particular language adopted for representing knowledge, and (ii) in the particular sort of inference rule adopted for reasoning.

Among the most known ones, it is certainly worth mentioning *propositional calculus*, *first-order logic*, *higher-order logic*, *modal logic*, up to *Horn clauses*, and *description logic*. Propositional calculus, for instance, adopts *modus ponens*, *modus tollens*, and *abduction* as inference rules, while other formalisms make different choices, e.g., *induction* or *modalities* (in modal logic). Horn clauses, in its turn, adopt the *resolution principle* (in several variants: SLD, SLDNF, tabled-SLD); the same applies to description logic.

## 2.2. Knowledge Representation

*Knowledge representation* (KR, Figure 1, top left) has always been regarded as a key issue since the early days of AI, since no reasoning can exist without knowledge. Its instances spread from *deductive data bases* [7] to *description logics* [8] and *ontologies* [9], to name just a few. Many kinds of logic-based knowledge representation systems have been proposed over the years, mostly relying on description logics and modal logics to represent, respectively, terminological knowledge and time-dependent or subjective knowledge.

Early KR formalisms, such as *semantic networks* and *frames* [10], also aimed at providing a structured representation of information. Description logics are based on the same ideas, but without the formal weaknesses that made the use of their precursors problematic. In DL, complex concepts are built from simpler ones with an emphasis on the decidability of the reasoning tasks and on the provision of sound, complete and empirically tractable reasoning services. Applications range from reasoning with database schemas and queries [11] to *ontology languages* such as OIL, DAML+OIL and OWL [12]—always keeping in mind that not only the key inference problems should be decidable, but also that the decision procedures should be implemented *efficiently*.

Ontology-based approaches are popular because of their basic goal—a common understanding of some domain that can be shared between people and application systems. At the same time, it should be understood that the general concepts and relations of a top-level ontology can rarely accommodate all the systems peculiarities [13,14].

Several systems based on DL have been developed—e.g., [15,16]—in diverse application domains, such as natural language processing, configuration of technical systems, software information systems, optimizing queries to databases, planning. It is especially worth mentioning Flora-2 [17], a rule-based, object-oriented knowledge base system for a variety of automated tasks on the *semantic web*, ranging from metadata management to information integration to intelligent agents. The system integrates F-logic, HiLog, and transaction logic into a coherent knowledge representation and inference language, resulting in an effective framework that combines the rule-based and the object-oriented paradigms.

### 2.3. Reasoning Approaches and Techniques

Logic-based reasoning approaches root back to John McCarthy's work of 1958 [18], aimed at developing the idea of formalizing the so-called *commonsense reasoning* to build *intelligent artifacts*. This also means to consider non-trivial involved issues, such as the need to formalize the situation, actions and causal laws, etc. Many tools have been developed over the years for commonsense reasoning formalization: there are freely available commonsense knowledge bases and natural language processing toolkits, supporting practical textual-reasoning tasks on real-world documents including analogy-making, and other context oriented inferences—see for instance [19–23]). There have been also several attempts to construct very large knowledge bases of commonsense knowledge by hand, one of the largest being the CYC program by Douglas Lenat at CyCorp [24].

The modern approach to automated theorem proving starts with Robinson's resolution principle [25]: since then, several technologies have exploited *deduction* on first-order logic knowledge base to provide reasoning capabilities in diverse areas—logic programming, deductive data bases, and constraint logic programming (CLP) possibly being the major ones. Other approaches and techniques, however, built upon the *induction* and *abduction* principles (Figure 1, middle left).

As its name suggests, *deduction* operates top-down, deriving a true conclusion from a universal true premise: logically speaking, this means that the conclusion's truth necessarily follows from the premise's truth. *Induction*, instead, operates bottom-up, basically making a guess—a generalization—from specific known facts: so, the reasoning involves an element of probability, as the conclusion is not based on universal premises. *Abduction* is somehow similar, but seeks for cause-effect relationships—i.e., the goal is to find out under which hypotheses (or premises) a certain goal is provable. Such technologies are exploited, in particular, for the verification of compliance of specific properties [26].

*Logic programming* (LP) is likely the most widely adopted technology based on deduction. From Colmerauer and Kowalsky's seminal work [27,28], the Prolog language has been since then one of the most exploited language in AI applications [29]. Other valuable approaches include *fuzzy logic*, *answer-set programming* (ASP), *constraint logic programming* (CLP), *non-monotonic reasoning*, and *belief-desire-intention* (BDI) (Figure 1, bottom left).

Fuzzy logic [30] aims at dealing with lack of precision or uncertainty. In this sense, it is perhaps closer in spirit to the human thinking than traditional logic systems. Not surprisingly, fuzzy approaches are exploited as a key technology in specific application areas, e.g., the selection of manufacturing technologies [31], and industrial processes where the control via conventional methods suffers from the lack of quantitative data about I/O relations. There, a fuzzy-logic controller effectively synthesizes an automatic control strategy from a linguistic control strategy based on an expert's knowledge.

*Answer-set programming* (ASP) and *constraint logic programming* (CLP) are the two main logical paradigms for dealing with various classes of NP-complete combinatorial problems. ASP solvers are aimed at computing the answer sets of standard logic programs; these tools can be seen as theorem provers, or model builders, enhanced with several built-in heuristics to guide the exploration of the solution space. Some of the best-known solvers are Clingo [32] and DLV [33].

Constraint logic programming (CLP) [34], perhaps the most natural extension of LP (or, its most relevant generalization), has evolved over the years into a powerful programming paradigm, widely used to model and solve hard real-life problems [35] in diverse application domains—from circuit verification to scheduling, resource allocation, timetabling, control systems, etc. CLP technologies can be seen as complementary to *operation research* (OR) techniques: while OR is often the only way to find the optimal solution, CLP provides generality, together with a high-level modeling environment, search control, compactness of the problem representation, constraint propagation, and fast methods to achieve a valuable solution [36]. CLP tools evolved from the ancestor—CHIP [37], the first to adopt constraint propagation—to the constraint-handling libraries of ILOG [38] and COSYTEC [39], up to CLP languages such as Prolog III [40], Prolog IV [41], CLP(R) [42], and clp(fd) [43].

Non-monotonic reasoning means to face the basic objection [44] that logic could not represent knowledge and commonsense reasoning as humans because the human reasoning is inherently *non-monotonic*—that is, consequences are not always preserved, in contrast to first-order logic. Since then, a family of approaches have been developed to suit specific needs—among these, *default reasoning* [45], *defeasible reasoning* [46], *abstract argumentation* theory [47]. Defeasible reasoning, in particular, is widely adopted in *AI and law* applications, to represent the complex intertwining of legal norms, often overlapping among each other, possibly from different, non-coherent sources. Abstract argumentation theory, in its turn, is concerned with the formalization and implementation of methods for rationally resolving disagreements, providing a general approach for modeling conflicts between arguments, and a semantics to establish if an argument can be acceptable or not.

Belief–desire–intention (BDI) logic is a kind of modal logic used for formalizing, validating, and designing cognitive *agents*—typically, in the *multi-agent systems* (MAS) context. A cognitive agent is an entity consisting of (i) a belief base storing the agent’s *beliefs*, i.e., what the agent knows about the world, itself, and other agents; (ii) a set of *desires* (or goals), i.e., the proprieties of the world the agent wants to eventually become true; (iii) a *plan* library, encapsulating the agent’s procedural knowledge (in the form of plans) aimed at making some goals become true; and (iv) a set of *intentions*, storing the states of the plans the agent is currently enacting as an attempt to satisfy some desires. All such data usually consist of first-order formulas. Then, the dynamic behavior of a BDI agent is driven by either internal (updates to the belief base or changes in the set of desires) or external (perceptions or messages coming from the outside) events, which may cause new intentions to be created, or current intentions to be dropped. By suitably capturing the revision of beliefs, and supporting the concurrent execution of goal-oriented computations, BDI architectures overcome critical issues of “classical” logic-based technologies—*concurrency* and *mutability*—in a sound way. Overall, BDI architecture leads to a clear and intuitive design, where the underlying BDI logic provides for the formal background. Among the frameworks rooted on a BDI approach, let us mention the AgentSpeak(L) [48] abstract language and its major implementation, namely Jason, Structured Circuit Semantics [49], Act Plan Interlingua [50], JACK [51], and dMARS—a platform for building complex, distributed, time-critical systems in C++ [52].

#### 2.4. Verification and Model-Checking

*Verification* (Figure 1, bottom left) is another crucial issue: in the logic context, it takes the form of *model-checking*—validating a (logic-based) model representation against some given (logic-based) specification. This process is typically performed by an automated tool, the *model checker*, which can be built adopting different philosophies—explicit/implicit-state model checkers, inductive model checkers, epistemic model checkers, and other.

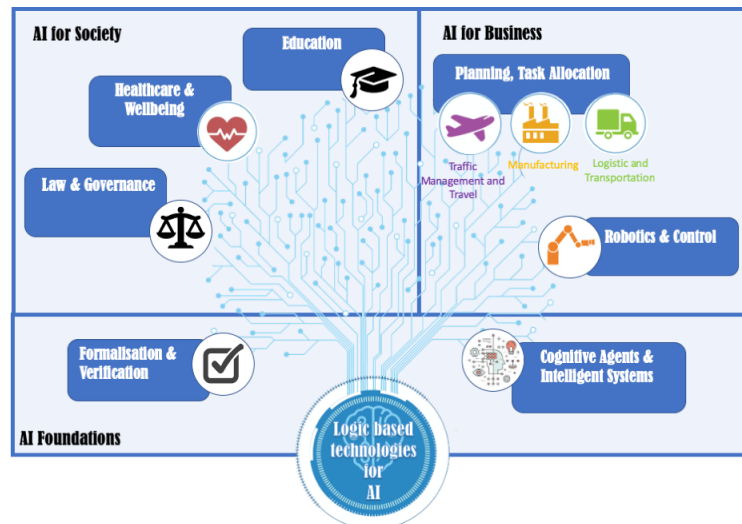
In short, *explicit-state* model checkers construct and store a representation of each visited state, while *implicit-state* model checkers adopt logical representations of sets of states (such as Binary Decision Diagrams) to describe the regions of the model state space that satisfy the properties under evaluation, thus achieving a more compact representation. *Inductive* model checkers exploit induction over the state transition relation to prove that a property holds over all the executable paths in a model, while *epistemic* model checkers focus on how the states of information of agents change over the time. The most recent symbolic approach is perhaps *SAT-based* (SAT-solvers are algorithms to solve the satisfiability problem for propositional formulas) model-checking, which translates the model-checking problem for a temporal-epistemic logic to a satisfiability problem of a formula in propositional logic, achieving higher efficiency.

Overall, model checkers have been used to verify, among the others, MAS and IoT systems, and are currently extensively used in the hardware industry, as well as to check (kinds of) software systems [53].



### 3. Logic-Based AI: Application Areas

The above techniques have been applied in a variety of fields: Figure 2 summarizes both the main application areas and their interconnections. For each application area, in the following we (i) introduce what the area is about, (ii) outline the main sub-categories (if any), (iii) discuss the role of logic in each sub-category as well as (iv) its benefits and limits, and (v) present the main actual applications.



**Figure 2.** Logic-based technologies application areas with respect to the main AI categories—namely AI Foundations, AI for Society, and AI for Business. Intentionally, the picture only illustrates the AI areas that are closely related to logic.

To make the comparison actually effective, Tables 1 and 2 summarize our findings from two different viewpoints. The first outlines the strengths of the diverse techniques per application area, i.e., where they provide the strongest contribution; the second puts each technique in relation to the different market segments, providing appropriate references to the literature where each technique is used.

**Table 1.** Sorts of logic per application area. Acronym and abbreviation key: FOL: First-Order Logic; DL: Description Logic; BDI: Belief–Desire–Intention (Logic); TL: Temporal Logic; FL: Fuzzy Logic; PL: Probabilistic Logic; DR: Defeasible Reasoning; CLP: Constraint Logic Programming.

	FOL	DL	BDI	TL	FL	PL	DR	CLP
Formalization & Verification				✓		✓		
Cognitive Agents	✓	✓	✓	✓	✓		✓	
Healthcare & Wellbeing	✓	✓				✓		
Law & Governance	✓						✓	
Education	✓	✓			✓			
Planning & Task Allocation	✓	✓					✓	✓
Robotics & Control		✓		✓	✓			✓

**Table 2.** Sorts of logic per market segment. Acronym and abbreviation key: FOL: First-Order Logic; DL: Description Logic; BDI: Belief-Desire-Intention (Logic); TL: Temporal Logic; PL: Probabilistic Logic; DR: Defeasible Reasoning; CLP: Constraint Logic Programming.

	FOL	DL	BDI	TL	FL	PL	DR	CLP
Aerospace					[54,55]			[56]
Analytics	[57]		[58]					
Bioinformatics	[57]	[12,17]			[59,60]	[61]		[56]
BPM	[57]	[17]						
Constructions								[56]
Critical systems				[62,63]				
CPS		[17]	[58,64]	[65]	[66,67]	[61]		
Cybersecurity					[68,69]	[61]		
Databases		[12]			[70,71]			
Decision support					[72,73]		[74]	
Energy		[24]	[58]	[65]	[75]	[61]		[56]
Finance		[24]			[76,77]			[56]
Government & Legal							[78]	
Hardware		[24]			[79,80]			[56]
Healthcare			[58,81]	[65]	[82,83]			
Information retrieval		[12]			[84,85]			
Logistic	[57]	[24]	[58,64]	[65]				[56]
Manufacturing			[58,64]	[65]	[86,87]			
Mechanics								[56]
Mobile applications				[88,89]				
Railways				[63]	[90,91]			
Telecommunications	[57]				[92,93]	[61]		[56]
Transports			[58]		[94,95]			
Web services		[12,17]		[96,97]	[98,99]			

### 3.1. AI Foundations

#### 3.1.1. Formalization and Verification of Computational Systems

Formalization and verification of computational systems refer to a collection of techniques for the automatic analysis of reactive systems—in particular, safety-critical systems, where subtle design errors can easily elude conventional simulation and testing techniques.

The main logic-based technology exploited in this field is model-checking—presently a standard procedure for quality assurance both because of its cost-effectiveness and of the ease of integration with more conventional design methods. The model checker input is a description of the system to be analyzed and several properties that are supposed to hold: logic is used both to formalize the system description—states, transitions, model description, and specifications to be verified—and to express the behavioral aspects, capturing the key properties of information flow. Accordingly, such descriptions are often expressed in temporal and probabilistic logic (and their extension/variations). The model checker input is then a (usually finite-state) description of the system to be analyzed, and the expected properties in terms of temporal-logic formulas.

Application examples (Section 2.4) are notably heterogeneous, since model-checking is multidisciplinary and cross-field by its very nature.

Model-checking can provide a significant increase in the level of confidence of a system, enabling system verification a-priori, a-posteriori, and—what is most relevant presently—at run-time. On the other hand, any validation is, by definition, only as good as the system model itself: so, the validation result strongly depends on the precision of the input model. In addition, model-checking can turn out to be unsuitable for data intensive applications, as it increases the number communications.

### 3.1.2. Cognitive Agents and Intelligent Systems

Cognitive architectures are design methodologies, i.e., collections of knowledge and strategies applied to the problem of creating *situated intelligence*. Here, the main technologies borrow from *multi-agent systems* (MAS), since the cognitive architecture can be considered to be the brain of an agent reasoning to solve problems, achieving goals and taking decisions.

Generally speaking, cognitive agents in intelligent MAS straightforwardly exploit the logic-based models and technologies for the rational process, knowledge representation, expressive communication, and effective coordination. Developing an agent means to set up a deduction process: each agent is encoded as a logic theory, and selecting an action means to perform a deduction that reduces the problem to a solution, as in theorem proving. Logic can also be used to represent the agent's surrounding *environment* and the *society* of agents—that is, overall, two of the three key aspects when it comes model the structure and dynamics of non-trivial MAS [100].

Technologies reflect the above context: many are related to agent programming and reasoning (Section 2.3), others to agent reliability and verification (Section 2.4). Many others focus on the societal aspect of cognitive architectures, by interpreting society as the ensemble where the collective behavior of the MAS are *coordinated* towards the achievement of the global system goals. Along this line, *coordination models* glue agents together by governing agent interaction, paving the way towards social intelligence: after the seminal work of Shared Prolog [101], notable examples are TuCSoN [102], ReSpecT [103], and AORTA [104].

Within an agent society, agents can enter into argumentation processes to reach agreements and dynamically adapt to changes: so, disputes and conflicts need to be managed in order to achieve a common agreement and establish the winner argument. Several technologies exist for solving reasoning tasks on the abstract argumentation frameworks [105]. Since problems of this kind are intractable, efficient algorithms and solvers are needed. As discussed in [106], most solvers are based on logic-based programming or logic frameworks including ASP and SAT-solvers.

Specific technologies exist for dealing with the *environment* abstraction of cognitive architectures, mostly in the *coordination* area. There, coordination/interaction artifacts work as run-time abstractions which encapsulate and support coordination/interaction services, usable as building blocks for designing and governing coordination, collaboration, competition services inside heterogeneous MASs. *Description spaces with fuzziness* [107] and *semantic tuple centers* [108] can be read as technologies for *situated interaction & coordination*, emphasizing the *situated* aspect of interaction, i.e., the environment-related aspect. In-between lies LPaaS (Logic Programming as a Service), a framework aimed at supporting the distribution of logic knowledge in the environment. There, artifacts work as knowledge repositories (in the form of environment structure and properties) while also embedding the reasoning process (enabled and constrained by the knowledge they embody).

When agents are immersed in a Knowledge-Intensive Environment (KIE), the cognition process goes beyond that of the individual agent, and distributed cognition processes may take place, promoting the idea of *intelligent environment* [109]. In such a way, the environment concept is extended beyond situated action—which, by the way, motivates the inclusion of the semantic web within the macro-area of environmental abstractions.

Moving from [110], the “Agents in the semantic web” sub-category lists JADL, AgentOWL, and EMERALD, which exploit semantic web technologies to inter-operate.



Hybrid cognitive architectures have recently gained attention to combine symbolic and sub-symbolic (emergent) approaches. Examples are ACT-R [111]—based on the modeling of human behavior—CLARION [112] and LIDA [113].

Despite the simplicity and elegance of the logical semantics in logic-based architectures, some issues do exist. The *transduction* problem [114] has to do with the difficulty of accurately translating the model into a symbolic representation, especially in a complex environment. One more difficulty comes from suitably representing information in such a symbolic form that agents can reason about, with and in a time-constrained environment. Finally, the transformation of percepts may not be accurate enough to describe the environment itself, due to sensor faults, reasoning errors, etc.

### 3.2. AI for Society

#### 3.2.1. Healthcare and Wellbeing

In the healthcare domain, AI typically takes the form of complex algorithms and software systems to emulate human cognition in the analysis of complicated medical data, approximating conclusions without direct human input. The primary aim here is to analyze relationships between prevention or treatment techniques and patient outcomes. AI programs have been developed and applied to practices such as diagnosis processes, treatment protocol development, drug development, personalized medicine, and patient monitoring and care.

In this field, logic is exploited to represent knowledge in a human-understandable way, and reason on it via properly formalized rules—in particular, decision support (symbolic) rules, obtained from domain experts and/or decision models induced from data.

At the same time, symbolic logic scales does not scale easily: knowledge engineers need to extract the logic by interviewing or observing human experts. On the other hand, sub-symbolic techniques such as supervised deep learning scale more easily, but are subject to bias in the training data—and, of course, their outcome cannot be explained.

Here again, the semantic web provides a technical framework for the formal semantic modeling—i.e., interpretation, abstraction, axiomatization, and annotation—of healthcare knowledge in terms of classes, properties, relations and axioms [115]. The semantic web framework for healthcare systems provide notable features: (i) semantic modeling of the procedural and declarative healthcare knowledge as ontologies, hence a semantically rich and executable knowledge representation formalism; (ii) annotation—typically via RDF (Resource Description Framework)—of healthcare knowledge artifacts guided by the ontological model of the knowledge artifact, so as to characterize the main concepts and relations within the artifact; (iii) representation of different patient data sources in a semantically enriched formalism that helps to integrate heterogeneous data sources by establishing semantic similarity between data elements; (iv) semantic interoperability between multiple ontologies, using ontology alignment and mediation methods to dynamically synthesize/shape multiple knowledge resources so as to address all the facets of the specific healthcare problem; (v) specification of the decision-making logic in terms of symbolic rules, which can be executed using proof engines to infer suitable recommendations/actions; and (vi) provision of a justification trace of the inferred recommendations, so as to let users understand the rationale of the recommended interventions [116].

Among the healthcare systems based on reasoning, CARA (Context-Aware Real-time Assistant) [83] aims at providing personalized healthcare services for chronic patients in a timely manner, adapting the healthcare technology so that it fits in both with the normal activities of the elderly and with the working practices of the caregivers. Based on a fuzzy-logic context model and a related context-aware reasoning middleware, CARA provides context-aware data fusion and representation, as well as inference mechanisms that support remote patient monitoring and caregiver notification.

### 3.2.2. Law and Governance

Due to its wide potential impact on society and economy, *AI and law* is one of today's most relevant research areas. It consists of an interdisciplinary effort combining methods and results from several sources, from deontic logic, norms and agent-based simulation to game theory and norms, normative agents, norms and organization, norms and trust, norms and argumentation.

Contributions in the field of AI and law are strongly connected with the aforementioned agent architectures. *Agreement technologies* [117], in particular, is a new vision outlining next-generation, open, distributed systems where interaction between computational agents could be based on the notion of *agreement*. This calls for (i) a normative context defining the rules of the game, or the "space" of agreements that the agents can possibly reach; (ii) an interaction mechanism to establish (first) and enact (then) agreements; and (iii) a joint research effort from several fields—including, but not limited to, multi-agent systems, semantic technologies, social sciences—aimed at fruitfully combining results and contributions from all such areas—like, for instance, semantic alignment, negotiation, argumentation, virtual organizations, learning, real time, and others. Semantic web standards provide a good basis for representing the knowledge of local agents, the functionalities and everything needed to achieve a goal in agreement with other agents.

However, the formalisms behind these technologies fall short when dealing with the distributed, open and heterogeneous nature of AT systems where agents may have different views of the world and, therefore, mutually inconsistent knowledge bases. To cope with this issue, new logical formalisms—specifically aimed at handling situations where pieces of knowledge are independently defined in different contexts—have been defined, extending classical logics in order to deal with incomplete and *defeasible* knowledge. Logic is thus exploited to represent the knowledge with a high degree of peculiarity (for instance defeasibility, but also the possibility to discern among permission, obligation or beliefs in deontic logic), and to reason over such knowledge.

Many interesting experiments have been performed in this application area. Notable defeasible argumentation implementations, aimed at supporting reasoning and resolve inconsistencies, are Defeasible Logic Programs [118], ASPIC [119], and ABA [120]. Several other applications use ontologies and legal search engines [121], which exploit advanced search technology from AI, data mining, data analytics, ontologies and natural language processing [122]. The main issues that remain unsolved in this area is that a unique and general framework for dealing with norms and argumentative issues is still missing: in fact, most solutions are too narrow in scope, tailored to specific use cases, other than being possibly weak from the software engineering perspective.

In short, logic-based approaches in the legal field [123] (i) help formalize legal norms and concepts in a clear and understandable way, thus enabling verification and the detection of unfair policies, or the violation of essential rights; (ii) support explanatory and arguable decisions in the regulatory context.

In general, however, current tools are unable to imitate advanced cognitive processes such as human reasoning, understanding, meta-cognition or the contextual perception of abstract concepts that are essential for legal thinking. Indeed, a lawyer's work is often very complex, implying the management and processing of huge amounts of data, where to find correlations between facts and circumstances, and formulate reasoned opinions and action guidelines taking into account all the applicable rights and obligations. This is why the process of understanding and formulating a decision is mostly creative—the result of a complex cognitive process.

### 3.2.3. Education

AI has been part of many e-learning platforms for a long time, with applications ranging from personalized learning, recommendation of resources, automated grading, to prediction of attrition rates—to name just a few. The rapid expansion of the *educational technology* industry is now further pushing and exploiting advanced AI-enabled learning technologies.

Within this area, symbolic AI techniques have been used in adaptive educational systems, such as fuzzy-logic, decision tree, etc. there, logic has been mainly applied for knowledge management and

recommendation. In some systems, for instance, the focus is on examining and assessing the student characteristics in order to generate students' profiles, to be used for evaluating their overall level of knowledge and, consequently, as a basis for prescribed software pedagogy. Symbolic AI approaches are used to support the diagnostic process, so that the course content can be adjusted to cater each student's needs. Some of them, in addition, are also used to learn from student's behavior to adjust the prescribed software pedagogy.

Applications are related to semantic web technologies, contextualized to e-learning so as to adapt instruction to the learner's cognitive requirements in three ways—background knowledge, knowledge objectives and the most suitable learning style [124,125]. Over the years, fuzzy-logic techniques and logic MAS have also been experimented for e-learning purposes. In particular, in [126,127], a fuzzy-logic-based system learns the users' preferred knowledge delivery to generate a personalized learning environment; whereas in [128] agents detect, recognize, eliminate, and repair the faults of the e-learning course, keeping the system up and working, providing robustness. Another interesting application in the domain of formal logic proofs, taken as the base of several further extensions, is the Logic-ITA (Intelligent Teaching Assistant) web-based system [129]: its purpose is to soothe the issues caused by large classes or distance learning, acting as an intermediary between teacher and students. On the one hand, it provides students with an environment to practice formal proofs, giving proper feedback; on the other, it allows teachers to monitor the class's progress and mistakes.

Although the impact on classrooms has been relatively minor so far, the potential of AI in education is high and likely to increase, as demonstrated by the many European actions/projects currently in place [130,131]. The main challenges and issues concern the creation of a sustainable educational environment, capable of developing equitable education even for the least developed countries—to be dealt with at the suitable political level.

### 3.3. AI for Business: Automation and Robotics

Automation is probably the earliest and perhaps most impactful application area for AI, as it represents the first step towards machine autonomy. Autonomy, in turn, is highly desirable whenever there is a need to re-design, re-build, or re-program machines while the deployment context evolves. *Autonomous* machines differ from *automatic* ones in that designers no longer need to forecast any possible situation, because the machine is programmed for learning or planning. This is particularly interesting in *cyber-physical systems* (CPS) [132] and *robotics*, where machines have a physical body that makes them capable of affecting (or being affected by) the physical world.

Needless to say, applications of automation and robotics in industry are manifold, and so are the corresponding research lines. In the following we explore the role and impact of logic-based paradigms and technologies in such areas.

#### 3.3.1. Planning and Task Allocation

Planning and scheduling are one of the oldest fields in AI, also related to multi-agents and cognitive architectures: research has mainly to do with the decision-making process that determines what, when, where, and how to reach a goal and compute a task. There, logic is exploited to represent the knowledge domain, its constraints, and the reasoning mechanism.

Logic-based scheduling methodologies include *rule-based* approaches and *constraint-guided* search. Rule-based scheduling methods aim to emulate the decision-making behavior of human schedulers, captured in terms of suitable logic rules. Correspondingly, rule-based systems are typically envisioned to replicate the actions of experienced humans with specific scheduling skills. Unsurprisingly, this is one of the most successful application domains of CLP techniques: the scheduler goal is to identify feasible solutions which balance different constraints or schedule requirements.

Applications spread from manufacturing to traffic scheduling and management (e.g., autonomous vehicles, aircraft, ...), up to urban search and rescue activities (e.g., traffic assignment in natural disaster evacuations, ...), and many others. A typical example of a constraint-based scheduling application

is ATLAS [133], which schedules the production of herbicides at the Monsanto plant in Antwerp. PLANE [134] is another system used at Dassault Aviation to plan the production of the military Mirage 2000 jet and the Falcon business jet: the objective is to minimize changes in the production rate, which has a high set-up cost, while finishing the aircraft just in time for delivery. The COBRA system [134] generates workplans for train drivers of North-Western Trains in the UK: each week, about 25,000 activities need to be scheduled in nearly 3000 diagrams on a complex route network. The DAYSY Esprit project [135] and the SAS-Pilot program [136] consider the operational re-assignment of airline crews to flights. The STP (Short Term Planning) application at Renault [137] assigns product orders to factories to minimize transportation costs. The MOSES application by COSYTEC [137] schedules the production of compound food for different animal species, eliminating contamination risks while satisfying customer's demand at the minimal cost. FORWARD [138] is a decision support system, based on CHIP, used in three oil refineries in Europe to tackle all the scheduling problems occurring in the process of crude oil arrival, processing, finished product blending and final delivery. Finally, Xerox has adopted a constraint-based system for scheduling various tasks in reprographic machines (like photocopiers, printers, fax machines, etc.); the scheduler is supposed to determine the sequence of print making and coordinate the time-sensitive activities of the several hardware modules that make up the machine configuration [139].

Overall, the CLP approach faces well some of the key issues such as development time, nodes visited in the search tree, number of generated feasible solutions, and efficiency. At the same time, the very nature of such systems mandates for considerable development and tuning effort for each new application, as there is no expert to emulate. More generally, the main two issues are (i) the lack of a structured way to carry the insight gained from one application to the next, and (ii) the complexity of generating the symbolic knowledge that fully describes the application domain.

### 3.3.2. Robotics and Control

Cognitive architectures, planning, and task allocation techniques have been widely applied to robotics and control system: indeed, robotics applications translate the agent abstraction of cognitive architecture into a mechanical robot capable of doing action and taking decision.

Logic in robotics is, much more than elsewhere, tailored to the specificity of the application field, since control mechanisms need to control the robot sensors and actuators, along with all their low-level control software (for instance, robot motion mandatorily requires a set of feedback control primitives in order to keep motion coherent). More generally, control systems are present in lifts, photocopiers, car engines, assembly lines, power stations, etc.

Again, the logic-based approaches (and technologies) that are mostly adopted in this context are CLP, fuzzy logic, and temporal logic: in fact, many works dealing with robotic reasoning [140] exploit languages and technologies detailed in Section 2.3. CLP-based applications are typically at the smaller end, where it is still possible to prove that some global properties can be guaranteed with a given control. Thus, CLP is often exploited to build control software for electro-mechanical systems with a finite number of inputs, outputs, and internal states: each component is connected to a small part of the overall system, so its behavior can be captured quite simply. However, when the system is considered to be a whole, the number of global states can become very large: this calls for a smart technology that is able to handle such combinatorial explosion [141].

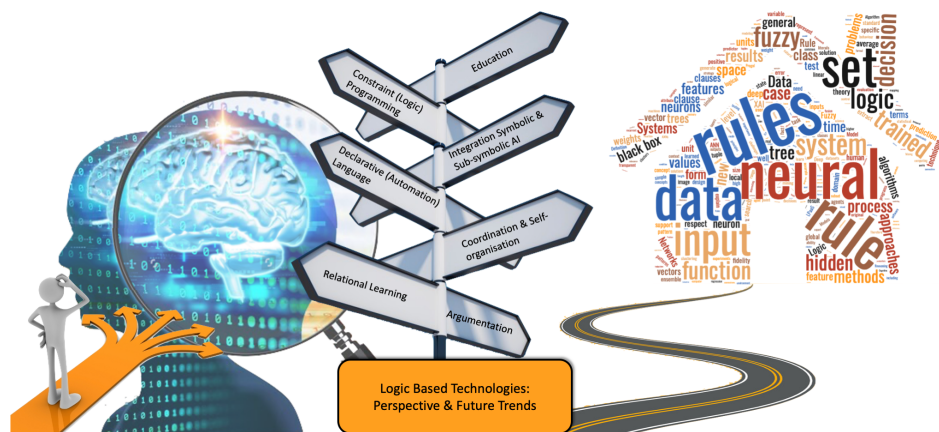
On the other side, many control systems have been developed exploiting a fuzzy-logic approach for dealing with real data which are sometimes imprecise, uncertain, complex, and with a high degree of randomness. Due to its good tolerance of uncertainty and imprecision, fuzzy logic has gained wide application in the area of advanced control of humanoid robots. For the same reason, fuzzy system are powerful tools to face crucial problems in industrial engineering and technology, such as risk management or product quality assurance, as well as in intelligent decision support system for adaptive industrial engineering [142–146]. Also, the hybrid techniques based on the integration of

neuro-fuzzy networks, neuro-genetic algorithms, and fuzzy-genetic algorithms are of great importance in the development of efficient algorithms [147].

Temporal-logic approaches and technologies have been exploited, e.g., for controlling robot motion or planning activities [148,149], because of their ability to reason over the time and its change, which makes it possible to build control laws to be verified over the time elapse. This is especially relevant for mobile robots, whose specifications are often temporal—even though time is not necessarily captured explicitly. For example, a swarm might be required to reach a certain position and shape eventually, or maintain a size smaller than a specified value until a final desired value is achieved. Other examples are collision avoidance among robots, obstacle avoidance, and cohesion, which are always required. In a surveillance mission, a selected area needs be visited “infinitely often” [150].

#### 4. Perspective and Future Trends

In this section, we systematize and frame the main perspectives for the evolution of logic-based technologies. To this end, Figure 3 means to illustrate and assess the state of the art discussed in Sections 2 and 3, in pair with the AI requirements mentioned in the Introduction. The main research directions for logic-based approaches and technologies are represented as directional signs, pointing at a cloud of words that aims to capture the main concepts behind logic-based technologies. The cloud is populated by the main keywords extracted from the papers cited in this section, taken as representatives of the most promising research lines. In particular, we focus on (i) promising *research areas*, as well as *applications*, leveraging on logic-based technologies, which we expect to grow in the near future; and (ii) promising research directions which involve logic-based technologies that are currently gaining momentum.



**Figure 3.** Perspective and future trends for logic-based technologies in symbolic AI. The directional signs illustrate the main research directions currently aiming to answer the AI’s hottest needs, sketched in the CoW (cloud of words) on the right. Words there come from the PDF of the papers discussed in Section 4—Perspectives and Future Trends.

##### 4.1. Integration of Symbolic and Sub-Symbolic AI

In the second half the 20th century—when AI was first recognized as a discipline by itself—several approaches towards machine intelligence became subject of intensive research efforts, leading to the vast corpus of literature and to the abundance of techniques available today.

Notably, two main families of approaches emerged—the *symbolic* and the *connectionist* or *sub-symbolic* ones [151,152]. While the former focuses on representing the world through symbols—which in turn, represent concepts—thus emulating how the human *mind* reasons and infers, the latter aim at mimicking human intuition by emulating how the human *brain* works at a very low level. Both families have both pros and cons, and stepped through both glory and misery—in terms of expectations, funding, research interest, and industry adoption [153,154]. Despite the effort devoted



to symbolic AI research and application in recent decades, numeric and connectionist approaches (e.g., neural networks) gained an unprecedented momentum since the early 2010s. However, even if it is currently not as popular as neural networks, the history of symbolic AI is extremely important as well—mostly because of the prominent influence on the many fields converging in AI: in fact, logic-based approaches represent the warhorse of symbolic AI.

In the recent years, the historical dichotomy between the “two souls” of AI has been reconciled, in favor of a comprehensive vision where symbolic and sub-symbolic approaches are seen as *complementary*—rather than in a competition—so that they mutually soften their corners [155–157] (see Figure 1). While symbolic approaches are well suited for relatively small-sized problems implying complex but exact tasks—possibly relying on structured data—sub-symbolic approaches are best suited to use cases processing big (possibly huge) amounts of possibly unstructured data—where errors, or lack of precision, are tolerated to some extent, if unavoidable. More precisely, complementarity between symbolic and sub-symbolic AI naturally emerges when comparing the two approaches under the following perspectives:

- sub-symbolic AI is *opaque*, meaning that human beings struggle in understanding the functioning and behavior of sub-symbolically intelligent systems; instead, symbolic AI is more *transparent*, as it is both human- and machine-interpretable at the same time
- sub-symbolic AI can improve itself *automatically* by consuming data, but it is difficult to extend and re-use outside the contexts it was designed for; conversely, symbolic AI is flexible and extensible, but requires humans to *manually* provide symbolic knowledge
- sub-symbolic AI is adequate for *fuzzy* problems where some (minimal) degree of error or uncertainty can be tolerated; whereas symbolic AI calls for precise data and queries provided by human beings, yet provides exact, *crisp* results as its outcome.

The two most active research lines that currently focus on unifying symbolic and sub-symbolic AI are *neural-symbolic computing* (NSC) [158] and *explainable artificial intelligence* (XAI) [1]. The former focuses on merging neural networks with logic-based technologies at a fundamental level, while the latter encompasses diverse studies aimed at exploiting symbolic AI to *explain* the internal functioning of sub-symbolic AI, thus making it more interpretable in the eyes of human beings. In the reminder of this section we explore both NSC and XAI.

#### 4.1.1. Techniques and Approaches: Hybrid Models for Intelligent Systems

Methods and approaches for the integration of symbolic and sub-symbolic techniques can be grouped under the heading of so-called “hybrid intelligent systems”, which exploit a combination of methods and techniques from AI sub-fields. The area includes several approaches such as—to name a few—neuro-fuzzy systems and hybrid connectionist-symbolic models, neural-symbolic computing, fuzzy and connectionist expert systems, evolutionary neural networks, genetic fuzzy systems, rough fuzzy hybridization, and Reinforcement Learning with fuzzy, neural, or evolutionary methods as well as symbolic reasoning methods. Generally speaking, hybrid approaches are based on the explicit integration of symbolic and sub-symbolic models, so as to take the best of each approach in matching the context needs. In the following, we discuss two of such approaches—namely neuro-fuzzy systems and neural-symbolic computing—as exemplars of the key points to be addressed in such integration: for a full discussion, we refer the reader to [159,160].

Neuro-fuzzy systems (NFS) [161–163] (also called fuzzy neural network) is a form of hybridization aimed at a synergy of neural and fuzzy systems, combining human-like reasoning with the learning approach of neural networks for providing “an effective vehicle for modeling the uncertainty in human reasoning” [164]. Knowledge is modeled as *if-then* rules, and the fuzzy nature provides the capability of universal approximation, somehow conjugating two contradictory requirements—interpretability vs. accuracy.

Neural-symbolic computing (NSC) combines the benefits of meta-heuristics, neural networks, and logic programming in order to incorporate two fundamental cognitive abilities—namely the ability

to *learn* and *reason* from what has been learned [165]—for diverse applications, such as finding an optimal solution in optimization problems [166,167]. NSC internalizes the benefits of novel and robust learning, and the logical reasoning and interpretability of symbolic simplification of *artificial neural networks* [168]: more recent advances in the fields of applied machine learning and reasoning can be found in [158,169]. The nature of integration, based on the convergence of neural processing with the symbolic representation of intelligence and reasoning, makes it particularly interesting for the development of explainable AI (XAI) systems, which focus on *interpretability* and *transparency*.

Indeed, the idea of logic as a programming language in neural networks to serve, represent and interpret a problem dates to [170]: the motivating force behind it is the idea that a single formalism is adequate both for logic and computing, and that it subsumes computation.

On the XAI wave, traditional logical approaches have in some sense been refurbished and are now experiencing a new dawn. However, a solid model is still to be defined, and a corresponding stable technology is still missing. For this reason, the field represents a hot issue at the frontier research level.

#### 4.1.2. Application Scenarios: Explainable, Responsible, and Ethical AI

Despite their large adoption, intelligent systems whose behavior is the result of ML-based procedures are difficult to trust for most people, in particular for non-experts. The reason is that numeric, sub-symbolic approaches tend to produce opaque systems, whose internal behavior is hard to explain even for their expert developers: opacity not only makes training and development error prone, but also prevents people from fully trusting—and thus accepting—the system itself. This is a key issue today in several contexts, as it is often not sufficient for intelligent systems to produce bare decisions—they must also be *explained*, as ethical and legal issues may arise. The financial and medical playgrounds are typical examples: according to current regulations, intelligible explanations need be output for each decision.

When it comes to build explainable ML-based systems, two are the main approaches. One consists of building systems which only leverage on transparent-by-construction ML predictors—like, for instance, decision trees. The other consists of leveraging on the full spectrum of ML techniques, trying to extract *post-hoc* symbolic explanations out of trained numeric predictors. The implicit strategy behind this approach is that symbols are far closer to what humans' conscious, rational mind is used to handle. The second case is of particular interest here as it aims at combining the best of the symbolic and sub-symbolic realms. Several works in recent decades proposed to extract symbolic knowledge from numeric models. As witnessed by several surveys [158,171–173] and works on the topic [174–184]—some of which are from the 80s or the 90s—the potential of symbolic knowledge *extraction* is well understood, although without hype.

At the current state of the art, a comprehensive and general framework tackling such a problem is still missing, as well as a technological implementation making the aforementioned techniques usable in practice. However, many research studies are exploiting the models and techniques mentioned in Section 4.1.1 to address the issue of explainable and responsible AI [185,186].

#### 4.2. Relational Learning

Most approaches to learning machines simply target specific activities where the task to be learned from data is both static and straightforward. For instance, despite the many applications of classification tasks and the plethora of things which can be recognized (a.k.a. properly classified) through well-engineered classifiers, one may hardly expect or require a ML-based classifiers to also *infer* the hierarchical relations among the classes it is able to recognize. Consider for example a deep neural network perfectly trained to recognize animals: it may easily discriminate among cats, dogs, and other mammals, but it would be very hard for it to learn what a mammal actually is. Similar concerns characterize other well-established tasks in ML such as regression, or best policy estimation.

*Statistical relational learning* (or just *relational learning* henceforth) is a research area, laying the intersection of logic, machine learning, and statistics, which essentially focuses on letting machines

learn *relations* among concepts, from examples. This is somewhat related to classification, except that relational learning is more general, as it is capable of *inferring* hierarchies of concepts from data, thus producing a symbolic knowledge base of *logic rules* as outcome, instead of a flat classifier.

The most common sort of inference performed by relational learning systems is *induction*. Roughly speaking, induction is the sort of reasoning performed by an agent willing to infer a *general* rule justifying several (positive or negative) examples or observations. As widely acknowledged in philosophy and logic, induction is very useful as it produces new valuable knowledge, but incomplete as the logic it produces is not *universal*. In other words, when leveraging on induction, one agent can never be 100% certain the rules it has inferred from data are (and will always be) true, but it may be more or less confident, depending on the statistics supporting the inferred rules, according to the available data. Thus, relational learning is inherently statistical in nature: the rules it induces are true up to a given probability.

Generally speaking, as far as future perspectives are concerned, relational learning is expected to bring about huge benefits in at least two areas: XAI and the semantic web.

As far as XAI is concerned, it is worth noting how relational learning is essentially a very intelligible and interpretable way to learn from data. Accordingly, it may have a role to play in the development of future explainable systems. Furthermore, by using it in combination with state-of-the-art image/language recognition technologies, relational learning may endow them with generalization facilities.

As far as the semantic web is concerned, it may have a role to play in the management of large and incomplete ontologies. In fact, semantic web technologies (such as RDF and OWL) can be exploited to build huge graphs of symbolic concepts and their relations, also known as ontologies. Such graphs are often built manually by communities of users or experts, or automatically by crawling other large sources of information such as Wikipedia. In both cases, the probability of inconsistencies, holes, or incompleteness to arise in the resulting knowledge graphs is high. In this framework, relational learning may have a role to play in complementing the aforementioned issues and an automatic and principled way.

#### 4.2.1. Inductive Logic Programming

*Inductive logic programming* (ILP henceforth) is arguably the most developed branch of *relational learning*—other than the soundest one, from a foundational point of view.

Several theoretical frameworks exist for ILP, as well as some logic-based technologies implementing them. For instance, the Progol technology leverages upon the entailment inversion framework [187]. Conversely, the CIGOL system leverages on resolution inversion [188], whereas the Golem system is based on the “relative least general generalization” approach [189]. Such technologies, have been exploited in many application scenarios, mostly involving the inference of rules and concepts out of logic knowledge bases. Despite their success, however, the technological development of ILP is still in its early stages. In fact, most ILP are currently unmaintained and poorly integrated with both mainstream logic- or ML-based technologies—even if ILP may have a role to play in both contexts.

Similar concerns arise for MAS as well. In fact, while the influence of computational logic in MAS is well established, and most agent- and logic-based programming frameworks (e.g., Jason [190]) are well integrated with *deductive* inference, a integration with ILP is still missing. Accordingly, that integration would be interesting as it would represent a straightforward means for learning symbolic knowledge.

#### 4.3. Constraint (Logic) Programming

*Constraint programming* (CP) is a programming paradigm for addressing search-related problems over particular domains—such as Boolean, as well as integer or real numbers). There, users are in charge of modeling a (satisfaction or optimization) problem in terms of *variables*—defined over the

mentioned domains—and the *constraints* they are subject to—which may be numerical equations or inequalities, as well as other relations, involving one, two or more variables at one time.

It is widely acknowledged how logic programming technologies—and, in particular, Prolog—can naturally represent both CP variables and constraints. For this reason, the CLP acronym soon emerged in the logic programming playground. It is used to indicate the field of *constraint logic programming* (CLP), i.e., the exploitation of logic programming as a linguistic and technological reification for CP.

CLP is considered the most natural generalization of logic programming—as LP is CLP over the Herbrand domain. Intuitively, the rationale behind such a shared opinion is that—differently from what happens for most search strategies, such as SLD—CLP does not commit to any particular order when exploring a search space. This of course paves the way towards the efficient exploration of infinite domains, other than the definition of purely declarative specifications of logic programs.

Despite several Prolog implementations—such as SICStus or SWI Prolog—have been extended to support logic programming over most common domains, CLP is not part of the standard ISO Prolog. Nevertheless, many Prolog implementations share a common syntax and semantics for their CLP modules, which often provide common functionalities. Thus, technological interoperability is not a major issue in this field.

However, when dealing with future perspectives, it would be interesting to extend CLP towards more domains than the classical ones. In particular, two main questions arising from this perspective are:

- will (C)LP ever reach full declarativeness? In other words, will it ever be possible to write CLP programs containing custom, user-defined domains and constraints?
- can sub-symbolic AI play a role in the development of more efficient or more expressive CLP solvers?

#### 4.4. Argumentation

*Argumentation* is part of the world around us: we are all increasingly interested in *why* a certain behavior occurs. This is actually the very reason behind the argumentative process: rather than just the mere reasoning, its ambition is to provide and document the *interaction* aspect—the dialogue that “subtitles” the reasoning. In the XAI perspective, getting to the root of certain behaviors and decisions becomes fundamental for intelligent systems: hence argumentation, with all its annexes, is an ineluctable research perspective. As discussed in Section 3.2.2, formal models of argumentation are making significant and increasing contributions to AI—from defining the semantics of logic programs, to implementing persuasive medical diagnostic systems, up to studying negotiation dialogues in multi-agent systems.

Therefore, from the two fundamental pillars of argumentation and AI, the perspectives in this area spread over the intersection of several fields such as autonomous agents, AI and law, logic in computer science, electronic governance, multi-agent systems. To be able to take part in an argumentative context, intelligent systems need to be explainable: here is why integrating symbolic and sub-symbolic approaches turns out to be, once again, a key theoretical and engineering issue. In turn, this calls for adequate supporting technologies—namely tools for argument analysis, evaluation, visualization, etc.

##### 4.4.1. Argumentation Mining

To this end, an intriguing research area is *argumentation mining* [191], which aims at automatically detecting, classifying and structuring argumentation in text: such a technology could become an important component of an argumentation analysis system, to understand the content of serial arguments, their linguistic structure, the relationship between the preceding and following arguments, and then recognize the underlying conceptual beliefs and finally understand within the comprehensive coherence of the specific topic.

One further intriguing perspective is to investigate the connection between such patterns and logical concepts such as ontologies: by mapping argumentation patterns onto existing suitable

ontologies on the matter of discourse, it could be possible to enable new kinds of reasoning, possibly building high-level aggregations aimed at synthesizing an argumentative process, to be possibly emulated. The ability to perform such sorts of analysis on these (possibly recurrent) structures in available data and texts opens a new range of challenging applications—for instance: verifying the compliance of a regulatory system; highlight injustices, discrimination, or prejudices; or even inject ethical (responsible, and explainable) behavior within the intelligent system.

#### 4.5. Coordination and Self-Organization

As discussed in Section 3.1.2, logic has been largely exploited for coordination in multi-agent systems—to represent and enact the coordination rules to govern/constrain the interaction space, express agent coordination and communication languages, represent the environment, etc. More generally, the logical formalization makes it possible to express coordination (specification, rules, etc.) in a rigorous yet readable way, both at design-time and run-time—and possibly analyze, verify, enact them. Indeed, the adoption of a logic-based representation creates the potential to formalize a “coordination theory” that could then be manipulated by specialized (meta-level) intelligent agents [192].

One of the major research perspectives regards the self-organization aspect. In fact, a logic-based approach can fruitfully and effectively support the increasing need for artificial systems to self-organize—which means self-configure, self-protect, possibly automatically recover from errors (self-healing), and self-optimize [193]. Going further, under the hypothesis to adopt a logic, tuple-based representation, one could devise a fully distributed, swarm-like strategy for clustering tuples according to their type—as a tuple template—so that tuples with same type are aggregated together. This could be particularly interesting when coupled with a (logic) tuple space abstraction (and infrastructure), which could work as the aggregator and natural support for subsequent reasonings by intelligent agents [194].

#### 4.6. Education

Educational institutions—from primary to higher education, as well as adult and professional learning—can benefit from the introduction of AI technologies into the learning process to actively support the achievement of learning objectives and enable new forms of learning patterns. One of the greatest challenges in this area concerns personalized education—that is, tailoring the path on the student’s learning skills, assisting educators with organizational tasks, etc. More in general, AI tools can play a role in improving the student-teacher relationship, e.g., for guaranteeing fair education especially in the developing countries, and overall in addressing the issues discussed in Section 3.2.3.

However, even more interesting than the “mere” integration of AI tools in education is the perspective of *AI education* itself: in order to build reliable, ethical, and fair applications, chances are that *AI itself* should be educated. This calls for contributions from the most diverse disciplines—from logic, to humanistic disciplines such as psychology, philosophy, ethics: the logic-based approach makes it possible not only to formalize shareable and human-understandable principles, but also to verify non-violation and compliance of AI with the principles themselves.

#### 4.7. Declarative Languages

One of the earliest and clearest advantages of logic-based technologies is declarativeness. In declarative programs one has simply to write *what* a machine should do, instead of *how* to do it. Declarativeness is today heavily involved in some very popular contexts—such as build/cloud automation languages (e.g., Gradle, OpenStack) or in continuous integration systems (e.g., Travis CI, GitLab)—even when a logic-based language is not used. Declarativeness in such contexts is often achieved via markup languages (e.g., XML, JSON or YAML) or a principled design of declarative API on top of functional languages (e.g., Gradle).



Consider for instance the case of OpenStack, an open-source technology aimed at supporting Infrastructure-as-a-Service (IaaS) Cloud providers. It is endowed with a module, namely Heat, aimed at supporting the orchestration of virtual resources for the Cloud users. From the user perspective, the usage of Heat is as simple as writing a Heat Orchestration Template (HOT) specification—that is, a *declarative* description of which virtual resources should be instantiated by Heat to set up the Cloud infrastructure. Declarativeness in that case is achieved through a JSON or YAML file describing virtual resources and their dependencies and relationships. The Heat engine is then capable of translating a well-formed HOT specification into an actual virtual infrastructure by considering dependencies and relationships among resources through an actual instantiation plan. A similar approach is exploited by Travis CI or GitLab, which still leverage on declarative YAML specification files to serve a different purpose. In this case, the goal is to let users automate the building, testing, or deploying phases of their software projects. However, despite the difference in their goals, such approaches share a common trait, i.e., a strong reliance on declarativeness attained through markup languages.

Of course, the practical exploitation of markup languages easily falls short when it comes to implement non-trivial, possibly parametric activities, involving loops, choices, string manipulation, or other computational constructs—like for instance instantiating a parametric number of resources to be differently customized according to some variable condition. There, implementors tend to tackle such issue through vendor-specific solutions for supporting loops or recursion, as well as conditional statements, or string manipulation through markup languages, thus giving birth to a plethora of poorly interoperable solutions.

To avoid such sort of issues, some technologies, such as Gradle, attempt to reach declarativeness by properly engineering their functional API, and by leveraging on mainstream languages with flexible syntaxes—such as Kotlin or Groovy. There, users usually declare the expected *tasks* to be performed on a software project. Inter-dependencies among tasks can be registered as well, along with execute-before or execute-after relations. To manage the project, in practice, users will then invoke some tasks among the ones previously defined (e.g., ‘compile’, ‘test’, or ‘deploy’) and let the Gradle system understand which tasks need to be executed for the requested task to become executable.

Despite such technologies are not logic-based in nature, their declarativeness is undeniable, as well as their success—as proven by their wide adoption. However, by comparing that sort of success against the moderate adoption of logic-based technologies, some questions immediately arise, which are to be answered by future research efforts in the field:

- what is favoring adoption of non-logic declarative technologies?
- what is preventing a wider adoption of logic-based declarative technologies in these areas?
- can computational logic and logic programming contribute in overcoming the current shortcomings of non-logic declarative technologies?

## 5. Discussion and Conclusions

As mentioned above, the new AI era calls for two fundamentals enabling factors: (i) the availability of big computing power even in minimal spaces, and (ii) the availability of huge amounts of context-related data. Such factors on the one side make it possible to learn from experience, which is the playground of sub-symbolic algorithms; on the other, make logic algorithms, historically used for expert systems and hence very effective as for transparency and explainability, computable *in good time*.

Big data have naturally led sub-symbolic approaches to prevail, because of their effectiveness in elaborating and getting valuable results from context-related data, learning trends and repeating patterns: yet, their inherent black-box nature is a clear issue. This is precisely where the integration with symbolic approaches can naturally provide added value, complementing symbolic and sub-symbolic approaches with each other. In fact, the two main weaknesses of symbolic approaches concern specifically (a) the extraction of context-related knowledge, and (b) computational complexity—the first being the natural territory of sub-symbolic approaches.

Computational complexity, in its turn, can be partially addressed thanks to the exponential increase of computational power, by suitably exploiting parallelism, as well as by re-defining and re-tuning symbolic approaches so as to fit presently computing paradigms and architectures—as in the case, for instance, of LPaaS [195]. However, it remains an issue in some application contexts, often leading to higher processing costs. An example is propositional interval temporal logics (ITL) [196], which provide a natural framework for representing and reasoning about temporal properties, but whose computational complexity constitutes a barrier for extensive use in practical applications. To cope with this issue, several approaches exploit constraints or adopt a locality principle; in other cases, as in DLs, complexity is decreased at the price of a limited expressiveness. For instance, in Bowman and Thompson’s decision procedure for propositional ITL, decidability is achieved by means of a simplifying hypothesis—the locality principle—that constrains the relation between the truth value of a formula over an interval and its truth values over initial sub-intervals [197]. Tableau-based decision procedures have been recently developed [198] for some interval temporal logics over specific classes of temporal structures, without resorting to any simplifying assumption.

Parallelism and concurrent programming techniques are another valuable tool to deal with complexity. The computing power of multi-level parallelism (MLP), in particular, constitutes a promising technique to facilitate concurrent programming while delivering performance comparable to that of fine-grained locking implementations—see for instance [199] and [200].

To recap, due to their strong foundations and features, logic-based technologies have the full potential to power symbolic approaches in such integration, opening intriguing perspectives currently under exploration in many research contexts, as discussed in Section 4. Tables 1 and 2, in particular, point out the very connections among the diverse logics, their application areas, and market segments. Table 1 highlights that cognitive agents and robotics are the application areas that exploit the widest spectrum of logic-based approaches, whereas other areas typically rely on a smaller subset. Reading the table orthogonally, first-order logic, description logic, and fuzzy logics appear to be the most general-purpose ones, the others being more specific. In its turn, Table 2 puts in evidence that some logics—such as description logic and fuzzy logics—are widely used in a variety of market segments, while the same does not hold for other logics, like BDI, defeasible reasoning, and probabilistic logic, thus opening a space of action for their possible expansion. BDI and defeasible reasoning, in particular, could effectively help to enable intelligent systems, intended as agents immersed in a continuous collaboration with humans, to behave and reason more “like a human”.

Overall, the synergy of symbolic and sub-symbolic approaches appears to be a viable and promising option to face key issues in today’s intelligent systems—namely the need for explainable, responsible, ethical AI. In particular, the adoption of symbolic approaches can help to achieve the key features of e-justice, fairness, ethics and transparency.

**Author Contributions:** The authors equally contributed to this work. All authors have read and agreed to the published version of the manuscript. Writing—original draft, R.C.; Investigation, G.C.; Methodology, E.D. and A.O. All authors have read and agreed to the published version of the manuscript.

**Funding:** One of the authors, Roberta Calegari, has been supported by project “CompuLaw”, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 833647).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gunning, D. *Explainable Artificial Intelligence (XAI)*; Funding Program DARPA–BAA–16–53; Defense Advanced Research Projects Agency (DARPA): Arlington, VA, USA, 2016.
2. Gallier, J.H. *Logic for Computer Science: Foundations of Automatic Theorem Proving*; Harper & Row Publishers, Inc.: New York, NY, USA, 1985.
3. Boyer, R.S.; Moore, J.S. *A Computational Logic*; Academic Press Professional, Inc.: Cambridge, MA, USA, 1988.

4. Levesque, H.J. A logic of implicit and explicit belief. In Proceedings of the 4th AAI Conference on Artificial Intelligence (AAAI '84); Number 5 in AAAI'84, Palo Alto, CA, USA, 6–10 August 1984; AAAI Press: Austin, TX, USA, 1984; pp. 198–202.
5. Omicini, A.; Denti, E.; Natali, A. Agent Coordination and Control through Logic Theories. In *Topics in Artificial Intelligence, Proceedings of the 4th Congress of the Italian Association for Artificial Intelligence (AI\*IA'95), Florence, Italy, 11–13 October 1995*; Gori, M., Soda, G., Eds.; LNAI; Springer: Berlin/Heidelberg, Germany, 1995; Volume 992, pp. 439–450. doi:10.1007/3-540-60437-5\_43. [[CrossRef](#)]
6. Bordini, R.H.; Braubach, L.; Dastani, M.; Seghrouchni, A.E.F.; Gomez-Sanz, J.J.; Leite, J.; O'Hare, G.; Pokahr, A.; Ricci, A. A survey of programming languages and platforms for multi-agent systems. *Informatica* **2006**, *30*, 33–44.
7. Green, C.C.; Raphael, B. The use of theorem-proving techniques in question-answering systems. In Proceedings of the 23rd ACM National Conference, Las Vegas, NV, USA, 27–29 August 1968; pp. 169–181. [[CrossRef](#)]
8. Baader, F. Basic description logics. In *The Description Logic Handbook: Theory, Implementation, and Applications*; Cambridge University Press: Cambridge, MA, USA, 2003; pp. 43–95.
9. Cimiano, P. *Ontology Learning and Population from Text*; Springer: New York, NY, USA, 2006.
10. Sowa, J.F. (Ed.) *Principles of Semantic Networks: Explorations in the Representation of Knowledge*; Morgan Kaufmann Series in Representation and Reasoning; Morgan Kaufmann Pub.: Burlington, MA, USA, 1991.
11. Artale, A.; Franconi, E.; Wolter, F.; Zakharyashev, M. A Temporal Description Logic for Reasoning over Conceptual Schemas and Queries. In Proceedings of the European Workshop on Logics in Artificial Intelligence (JELIA 2002), Cosenza, Italy, 23–26 September 2002; Springer: Cham, Switzerland, 2002; pp. 98–110. [[CrossRef](#)]
12. Horrocks, I. OWL: A Description Logic Based Ontology Language. In *Principles and Practice of Constraint Programming (CP 2005)*; Van Beek, P., Ed.; Extended Abstract; Springer: Cham, Switzerland, 2005; pp. 5–8. doi:10.1007/11564751\_2. [[CrossRef](#)]
13. Van Engers, T.; Boer, A.; Breuker, J.; Valente, A.; Winkels, R. Ontologies in the legal domain. In *Digital Government*; Springer: Cham, Switzerland, 2008; pp. 233–261.
14. Valente, A. Types and roles of legal ontologies. In *Law and the Semantic Web*; Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A., Eds.; Springer: Cham, Switzerland, 2005; pp. 65–76. [[CrossRef](#)]
15. Cohen, W.W.; Hirsh, H. Learning the CLASSIC description logic: Theoretical and experimental results. In *Principles of Knowledge Representation and Reasoning*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 121–133.
16. Moller, R.; Haarslev, V. Description logic systems. In *The Description Logic Handbook: Theory, Implementation, and Applications*; Cambridge University Press: Cambridge, MA, USA, 2003; pp. 282–305.
17. Yang, G.; Kifer, M.; Zhao, C. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *Lecture Notes in Computer Science, Proceedings of the OTM Confederated International Conferences On the Move to Meaningful Internet Systems, Catania, Italy, 3–7 November 2003*; Meersman, R., Tari, Z.S.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2888; pp. 671–688.
18. McCarthy, J.; Shannon, C. Automata Studies. *J. Symb. Log.* **1958**, *23*, 59–60. [[CrossRef](#)]
19. Lieberman, H.; Liu, H.; Singh, P.; Barry, B. Beating common sense into interactive applications. *AI Mag.* **2004**, *25*, 63.
20. Trinh, T.H.; Le, Q.V. A Simple Method for Commonsense Reasoning. *arXiv* **2018**, *arXiv:1806.02847*.
21. Liu, H.; Singh, P. ConceptNet—A practical commonsense reasoning tool-kit. *BT Technol. J.* **2004**, *22*, 211–226. [[CrossRef](#)]
22. Liu, H.; Lieberman, H.; Selker, T. GOOSE: A goal-oriented search engine with commonsense. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain, 29–30 May 2002*; De Bra, P., Brusilovsky, P.C.R., Eds.; Springer: Cham, Switzerland, 2002; Volume 2347, pp. 253–263.
23. Shapiro, S.C. SNePS: A logic for natural language understanding and commonsense reasoning. In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*; MIT Press: Cambridge, MA, USA, 2000; pp. 175–195.

24. Lenat, D.B. CYC: A large-scale investment in knowledge infrastructure. *Commun. ACM* **1995**, *38*, 33–38. [[CrossRef](#)]
25. Robinson, J.A. A machine-oriented logic based on the resolution principle. *J. ACM* **1965**, *12*, 23–41. [[CrossRef](#)]
26. Montali, M.; Torroni, P.; Chesani, F.; Mello, P.; Alberti, M.; Lamma, E. Abductive logic programming as an effective technology for the static verification of declarative business processes. *Fundam. Informaticae* **2010**, *102*, 325–361. [[CrossRef](#)]
27. Kowalski, R. Predicate logic as programming language. In Proceedings of the IFIP Congress, Stockholm, Sweden, 5–10 August 1974; pp. 569–544.
28. Colmerauer, A. Theoretical model of Prolog II. In *Logic Programming and its Applications*; van Canegham, M., Warren, D.H., Eds.; Ablex Publishing Corporation: New York, NY, USA, 1986; pp. 3–31.
29. Dawson, S.; Ramakrishnan, C.R.; Warren, D.S. Practical Program Analysis Using General Purpose Logic Programming Systems—A Case Study. In Proceedings of the ACM SIGPLAN 1996 Conference on Programming Language Design and Implementation, Philadelphia, PA, USA, 21–24 May 1996; PLDI '96; Association for Computing Machinery: New York, NY, USA, 1996; pp. 117–126, doi:10.1145/231379.231399. [[CrossRef](#)]
30. Yen, J.; Langari, R. *Fuzzy Logic: Intelligence, Control, and Information*; Prentice Hall Press: Upper Saddle River, NJ, USA, 1999; Volume 1.
31. Goyal, S.; Grover, S. Applying fuzzy grey relational analysis for ranking the advanced manufacturing systems. *Grey Syst. Theory Appl.* **2012**, *2*, 284–298. [[CrossRef](#)]
32. Gebser, M.; Kaminski, R.; Kaufmann, B.; Schaub, T. Clingo = ASP + Control: Preliminary Report. In Proceedings of the Technical Communications of the 30th International Conference on Logic Programming (ICLP'14), Vienna, Austria, 19–22 July 2014.
33. Eiter, T.; Faber, W.; Leone, N.; Pfeifer, G. Declarative problem-solving using the DLV system. In *Logic-Based Artificial Intelligence*; Springer: Boston, MA, USA, 2000; pp. 79–103, doi:10.1007/978-1-4615-1567-8\_4. [[CrossRef](#)]
34. Jaffar, J.; Lassez, J.L. Constraint logic programming. In Proceedings of the 14th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, Munich, Germany, 21–23 January 1987; pp. 111–119, doi:10.1145/41625.41635. [[CrossRef](#)]
35. Rossi, F. Constraint (Logic) Programming: A Survey on Research and Applications. In *New Trends in Constraints*; Apt, K.R., Monfroy, E., Kakas, A.C., Rossi, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 40–74.
36. Rossi, F.; Van Beek, P.; Walsh, T. Constraint programming. *Found. Artif. Intell.* **2008**, *3*, 181–211.
37. Simonis, H. The CHIP system and its applications. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Principles and Practice of Constraint Programming, Cassis, France, 19–22 September 1995*; Montanari, U., Rossi, F., Eds.; Springer: Berlin/Heidelberg, Germany, 1995; Volume 976, pp. 643–646.
38. Cplex, I.L.O.G. *11.0 User's Manual*; ILOG SA: Gentilly, France, 2007; pp. 32.
39. Aggoun, A.; Beldiceanu, N. Extending CHIP in order to solve complex scheduling and placement problems. *Math. Comput. Model.* **1993**, *17*, 57–73. [[CrossRef](#)]
40. Colmerauer, A. An introduction to Prolog III. In *Computational Logic*; Springer: Cham, Switzerland, 1990; pp. 37–79.
41. Benhamou, F.; Touraïvane, T. Prolog IV: Langage et algorithmes. In Proceedings of the JFPLC'95, IVèmes Journées Francophones de Programmation en Logique & Journée d'étude Programmation par Contraintes et Applications Industrielles, Dijon, France, 17–19 May 1995; pp. 51–64.
42. Jaffar, J.; Michaylov, S.; Stuckey, P.J.; Yap, R.H.C. The CLP(R) language and system. *ACM Trans. Program. Lang. Syst.* **1992**, *14*, 339–395. [[CrossRef](#)]
43. Codognet, P.; Diaz, D. Compiling constraints in clp (FD). *J. Log. Program.* **1996**, *27*, 185–226. [[CrossRef](#)]
44. Minsky, M. A Framework for Representing Knowledge Representation. In *The Psychology of Computer Vision*; Mc Graw-Hill: New York, NY, USA, 1975.
45. Reiter, R. A logic for default reasoning. *Artif. Intell.* **1980**, *13*, 81–132. [[CrossRef](#)]
46. Pollock, J.L. Defeasible reasoning. *Cogn. Sci.* **1987**, *11*, 481–518. [[CrossRef](#)]
47. Bondarenko, A.; Dung, P.M.; Kowalski, R.A.; Toni, F. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.* **1997**, *93*, 63–101. [[CrossRef](#)]

48. Rao, A.S. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In *Lecture Notes in Computer Science, Agents Breaking Away, Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, 22–25 January 1996*; de Velde, W.V., Perram, J.W., Eds.; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1038, pp. 42–55, doi:10.1007/BFb0031845. [[CrossRef](#)]
49. Lee, J.; Durfee, E.H. Structured circuit semantics for reactive plan execution systems. In *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, 31 July–4 August 1994*; Hayes-Roth, B., Korf, R.E., Eds.; AAAI Press/The MIT Press: Cambridge, MA, USA, 1994; Volume 2, pp. 1232–1237.
50. Huber, M.J. JAM: A BDI-theoretic mobile agent architecture. In *Proceedings of the Third Annual Conference on Autonomous Agents, Seattle, WA, USA, 1–5 May 1999*; pp. 236–243.
51. Howden, N.; Rönquist, R.; Hodgson, A.; Lucas, A. Intelligent agents—summary of an agent infrastructure. In *Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, 28 May–1 June 2001*.
52. d’Inverno, M.; Kinney, D.; Luck, M. Interaction protocols in Agentis. In *Proceedings of the International Conference on Multi Agent Systems (Cat. No. 98EX160), Paris, France, 3–7 July 1998*; pp. 112–119.
53. Grumberg, O.; Veith, H. *25 Years of Model Checking: History, Achievements, Perspectives*; Springer: Cham, Switzerland, 2008; Volume 5000.
54. Castillo, O.; Cervantes, L.; Soria, J.; Sanchez, M.; Castro, J.R. A generalized type-2 fuzzy granular approach with applications to aerospace. *Inf. Sci.* **2016**, *354*, 165–177. [[CrossRef](#)]
55. Ooi, M.E.; Sayuti, M.; Sarhan, A.A.D. Fuzzy logic-based approach to investigate the novel uses of nano suspended lubrication in precise machining of aerospace AL tempered grade 6061. *J. Clean. Prod.* **2015**, *89*, 286–295. [[CrossRef](#)]
56. Jaffar, J.; Maher, M.J. Constraint Logic Programming: A Survey. *J. Logic Program.* **1994**, *19/20*, 503–581. [[CrossRef](#)]
57. Carlsson, M.; Mildner, P. SICStus Prolog—The first 25 years. *Theory Pract. Log. Program.* **2012**, *12*, 35–66. [[CrossRef](#)]
58. Leitão, P.; Karnouskos, S.; Ribeiro, L.; Lee, J.; Strasser, T.I.; Colombo, A.W. Smart Agents in Industrial Cyber-Physical Systems. *Proc. IEEE* **2016**, *104*, 1086–1101. [[CrossRef](#)]
59. Torres, A.; Nieto, J.J. Fuzzy logic in medicine and bioinformatics. *BioMed Res. Int.* **2006**, *2006*, 7. [[CrossRef](#)] [[PubMed](#)]
60. Xu, D.; Keller, J.; Popescu, M.; Bondugula, R. *Applications of Fuzzy Logic in Bioinformatics*; World Scientific: Singapore, 2008; Volume 9.
61. Kwiatkowska, M.Z.; Norman, G.; Parker, D. PRISM: Probabilistic Symbolic Model Checker. In *Lecture Notes in Computer Science, Proceedings of the Computer Performance Evaluation, Modelling Techniques and Tools 12th International Conference, TOOLS 2002, London, UK, 14–17 April 2002*; Field, T., Harrison, P.G., Bradley, J.T., Harder, U., Eds.; Springer: Cham, Switzerland, 2002; Volume 2324, pp. 200–204, doi:10.1007/3-540-46029-2\_13. [[CrossRef](#)]
62. Pasareanu, C.S.; Visser, W.; Bushnell, D.H.; Geldenhuys, J.; Mehltitz, P.C.; Rungta, N. Symbolic PathFinder: Integrating symbolic execution with model checking for Java bytecode analysis. *Autom. Softw. Eng.* **2013**, *20*, 391–425. [[CrossRef](#)]
63. Wu, H. Industrial Applications of Probabilistic Model Checking- A Model-based Approach for Embedded Networked Systems and Concurrent Data Structures. Ph.D. Thesis, RWTH Aachen University, Aachen, Germany, 4 July 2017.
64. Winikoff, M. JACK™ Intelligent Agents: An Industrial Strength Platform. In *Multi-Agent Programming: Languages, Platforms and Applications*; Bordini, R.H., Dastani, M., Dix, J., Fallah-Seghrouchni, A.E., Eds.; Springer: Cham, Switzerland, 2005; Volume 15, pp. 175–193.
65. Holzmann, G.J. The Model Checker SPIN. *IEEE Trans. Softw. Eng.* **1997**, *23*, 279–295. [[CrossRef](#)]
66. Alam, K.M.; El Saddik, A. C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access* **2017**, *5*, 2050–2062. [[CrossRef](#)]
67. Mbuli, J.; Trentesaux, D.; Clarhaut, J.; Branger, G. Decision support in condition-based maintenance of a fleet of cyber-physical systems: A fuzzy logic approach. In *Proceedings of the Intelligent Systems Conference (IntelliSys), London, UK, 7–8 September 2017*; pp. 82–89.
68. Goztepe, K. Designing fuzzy rule based expert system for cyber security. *Int. J. Inf. Secur. Sci.* **2012**, *1*, 13–19.



69. Linda, O.; Manic, M.; Vollmer, T.; Wright, J. Fuzzy logic based anomaly detection for embedded network security cyber sensor. In Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Paris, France, 11–15 April 2011; pp. 202–209.
70. Kacprzyk, J.; Yager, R.R.; Zadrozny, S. A fuzzy logic based approach to linguistic summaries of databases. *Int. J. Appl. Math. Comput. Sci.* **2000**, *10*, 813–834.
71. Kacprzyk, J.; Zadrozny, S.; Ziolkowski, A. FQUERY III+: A “human-consistent” database querying system based on fuzzy logic with linguistic quantifiers. *Inf. Syst.* **1989**, *14*, 443–453. [[CrossRef](#)]
72. Liu, K.F.R.; Lai, J.H. Decision-support for environmental impact assessment: A hybrid approach using fuzzy logic and fuzzy analytic network process. *Expert Syst. Appl.* **2009**, *36*, 5119–5136. [[CrossRef](#)]
73. Adriaenssens, V.; De Baets, B.; Goethals, P.L.M.; De Pauw, N. Fuzzy rule-based models for decision support in ecosystem management. *Sci. Total Environ.* **2004**, *319*, 1–12. [[CrossRef](#)]
74. Gómez, S.A.; Chesñevar, C.I.; Simari, G.R. ONTOarg: A decision support framework for ontology integration based on argumentation. *Expert Syst. Appl.* **2013**, *40*, 1858–1870. [[CrossRef](#)]
75. Suganthi, L.; Iniyar, S.; Samuel, A.A. Applications of fuzzy logic in renewable energy systems—A review. *Renew. Sustain. Energy Rev.* **2015**, *48*, 585–607. [[CrossRef](#)]
76. Bojadziev, G. *Fuzzy Logic for Business, Finance, and Management*; World Scientific: Singapore, 2007; Volume 23.
77. Gil-Lafuente, A.M. *Fuzzy Logic in Financial Analysis*; Springer: Cham, Switzerland, 2005.
78. Prakken, H. *Logical Tools for Modelling Legal Argument: A Study of Defeasible Reasoning in Law*; Springer Science & Business Media: Berlin, Germany, 2013; Volume 32.
79. Costa, A.; De Gloria, A.; Faraboschi, P.; Pagni, A.; Rizzotto, G. Hardware solutions for fuzzy control. *Proc. IEEE* **1995**, *83*, 422–434. [[CrossRef](#)]
80. Bolognani, S.; Zigliotto, M. Hardware and software effective configurations for multi-input fuzzy logic controllers. *IEEE Trans. Fuzzy Syst.* **1998**, *6*, 173–179. [[CrossRef](#)]
81. Croatti, A.; Montagna, S.; Ricci, A.; Gamberini, E.; Albarello, V.; Agnoletti, V. BDI personal medical assistant agents: The case of trauma tracking and alerting. *Artif. Intell. Med.* **2019**, *96*, 187–197. [[CrossRef](#)]
82. Abbod, M.F.; von Keyserlingk, D.G.; Linkens, D.A.; Mahfouf, M. Survey of utilisation of fuzzy technology in medicine and healthcare. *Fuzzy Sets Syst.* **2001**, *120*, 331–349. [[CrossRef](#)]
83. Yuan, B.; Herbert, J. Fuzzy CARA—A Fuzzy-Based Context Reasoning System For Pervasive Healthcare. *Procedia Comput. Sci.* **2012**, *10*, 357–365. [[CrossRef](#)]
84. Gupta, Y.; Saini, A.; Saxena, A.K. A new fuzzy logic based ranking function for efficient information retrieval system. *Expert Syst. Appl.* **2015**, *42*, 1223–1234. [[CrossRef](#)]
85. Horng, Y.J.; Chen, S.M.; Chang, Y.C.; Lee, C.H. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 216–228. [[CrossRef](#)]
86. Azadegan, A.; Porobic, L.; Ghazinoory, S.; Samouei, P.; Kheirkhah, A.S. Fuzzy logic in manufacturing: A review of literature and a specialized application. *Int. J. Prod. Econ.* **2011**, *132*, 258–270. [[CrossRef](#)]
87. Lin, J.L.; Lin, C.L. The use of grey-fuzzy logic for the optimization of the manufacturing process. *J. Mater. Process. Technol.* **2005**, *160*, 9–14. [[CrossRef](#)]
88. Kohan, A.; Yamamoto, M.; Artho, C.; Yamagata, Y.; Ma, L.; Hagiya, M.; Tanabe, Y. Java Pathfinder on Android Devices. *ACM SIGSOFT Softw. Eng. Notes* **2016**, *41*, 1–5. [[CrossRef](#)]
89. Martinelli, F.; Mercaldo, F.; Nardone, V. Identifying Insecure Features in Android Applications using Model Checking. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Funchal, Madeira, Portugal, 22–24 January 2018; Mori, P., Furnell, S., Camp, O., Eds.; SciTePress: Setubal, Portugal, 2018; pp. 589–596, doi:10.5220/0006758105890596. [[CrossRef](#)]
90. Sezer, S.; Atalay, A.E. Dynamic modeling and fuzzy logic control of vibrations of a railway vehicle for different track irregularities. *Simul. Model. Pract. Theory* **2011**, *19*, 1873–1894. [[CrossRef](#)]
91. Skarlatos, D.; Karakasis, K.; Trochidis, A. Railway wheel fault diagnosis using a fuzzy-logic method. *Appl. Acoust.* **2004**, *65*, 951–966. [[CrossRef](#)]
92. Ghosh, S.; Razouqi, Q.; Schumacher, H.J.; Celmins, A. A survey of recent advances in fuzzy logic in telecommunications networks and new challenges. *IEEE Trans. Fuzzy Syst.* **1998**, *6*, 443–447. [[CrossRef](#)]
93. Catania, V.; Ficili, G.; Palazzo, S.; Panno, D. Using fuzzy logic in ATM source traffic control: Lessons and perspectives. *IEEE Commun. Mag.* **1996**, *34*, 70–74. [[CrossRef](#)]
94. Sarkar, A.; Sahoo, G.; Sahoo, U.C. Application of fuzzy logic in transport planning. *Int. J. Soft Comput.* **2012**, *3*, 1. [[CrossRef](#)]

95. Quddus, M.A.; Noland, R.B.; Ochieng, W.Y. A high accuracy fuzzy logic based map matching algorithm for road transport. *J. Intell. Transp. Syst.* **2006**, *10*, 103–115. [[CrossRef](#)]
96. Donini, F.M.; Mongiello, M.; Ruta, M.; Totaro, R. A Model Checking-based Method for Verifying Web Application Design. *Electron. Notes Theor. Comput. Sci.* **2006**, *151*, 19–32. [[CrossRef](#)]
97. Artzi, S.; Kiezun, A.; Dolby, J.; Tip, F.; Dig, D.; Paradkar, A.M.; Ernst, M.D. Finding Bugs in Web Applications Using Dynamic Test Generation and Explicit-State Model Checking. *IEEE Trans. Softw. Eng.* **2010**, *36*, 474–494. [[CrossRef](#)]
98. Tran, V.X.; Tsuji, H. QoS based ranking for web services: Fuzzy approaches. In Proceedings of the 4th International Conference on Next Generation Web Services Practices, Seoul, Korea, 20–22 October 2008; pp. 77–82.
99. Chao, K.M.; Younas, M.; Lo, C.C.; Tan, T.H. Fuzzy matchmaking for web services. In Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers), Taipei, Taiwan, 28–30 March 2005; Volume 2, pp. 721–726.
100. Omicini, A. SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems. In *Agent-Oriented Software Engineering, Proceedings of the 1st International Workshop (AOSE 2000), Limerick, Ireland, 10 June 2000*; Ciancarini, P., Wooldridge, M.J., Eds.; Revised Papers; Lecture Notes in Computer Science; Revised Papers, Springer: Berlin/Heidelberg, Germany, 2001; Volume 1957, pp. 185–193 doi:10.1007/3-540-44564-1\_12. [[CrossRef](#)]
101. Ciancarini, P. Distributed Programming with Logic Tuple Spaces. *New Gener. Comput.* **1994**, *12*, 251–283. [[CrossRef](#)]
102. Omicini, A.; Zambonelli, F. Coordination for Internet Application Development. *Auton. Agents Multi-Agent Syst.* **1999**, *2*, 251–269, doi:10.1023/A:1010060322135. [[CrossRef](#)]
103. Omicini, A.; Denti, E. From Tuple Spaces to Tuple Centres. *Sci. Comput. Program.* **2001**, *41*, 277–294. [[CrossRef](#)]
104. Jensen, A.S.; Dignum, V.; Villadsen, J. The AORTA Architecture: Integrating Organizational Reasoning in Jason. In *Engineering Multi-Agent Systems*; Dalpiaz, F., Dix, J., van Riemsdijk, M.B., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8758, pp. 127–145.
105. Dung, P.M. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-person Games. *Artif. Intell.* **1995**, *77*, 321–357. [[CrossRef](#)]
106. Gaggli, S.A.; Linsbichler, T.; Maratea, M.; Woltran, S. Summary Report of the Second International Competition on Computational Models of Argumentation. *AI Mag.* **2018**, *39*, 77–79. [[CrossRef](#)]
107. Nardini, E.; Omicini, A.; Viroli, M. Description Spaces with Fuzziness. In Proceedings of the 26th Annual ACM Symposium on Applied Computing (SAC 2011), TaiChung, Taiwan, 21–25 March 2011; Palakal, M.J., Hung, C.C., Chu, W., Wong, W.E., Eds.; Volume II: Artificial Intelligence & Agents, Information Systems, and Software Development; 2011; pp. 869–876. [[CrossRef](#)]
108. Nardini, E.; Viroli, M.; Panzavolta, E. Coordination in open and dynamic environments with TuCSoN semantic tuple centres. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, 22–26 March 2010; Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C., Eds.; ACM: Sierre, Switzerland, 2010; pp. 2037–2044, doi:10.1145/1774088.1774515. [[CrossRef](#)]
109. Hollan, J.; Hutchins, E.; Kirsh, D. Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. *ACM Trans. Comput.-Hum. Interact. (TOCHI)* **2000**, *7*, 174–196. [[CrossRef](#)]
110. Hendler, J. Agents and the Semantic Web. *IEEE Intell. Syst.* **2001**, *16*, 30–37. [[CrossRef](#)]
111. Anderson, J.R.; Lebiere, C. The Newell Test for a Theory of Cognition. *Behav. Brain Sci.* **2003**, *26*, 587–601. [[CrossRef](#)]
112. Sun, R. (Ed.) The CLARION Cognitive Architecture: Extending Cognitive Modeling to Social Simulation. In *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*; Sun, R., Ed.; Cambridge University Press: Cambridge, UK, 2005; pp. 79–100, doi:10.1017/CBO9780511610721.005. [[CrossRef](#)]
113. Franklin, S.; Madl, T.; D’Mello, S.; Snaider, J. LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Trans. Auton. Ment. Dev.* **2014**, *6*, 19–41. [[CrossRef](#)]
114. Bie, T.D.; Cristianini, N. Convex methods for transduction. In *Advances in Neural Information Processing Systems*; Thrun, S., Saul, L.K., Schölkopf, B., Eds.; MIT Press: Cambridge, MA, USA, 2004; pp. 73–80.
115. Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic web. *Sci. Am.* **2001**, *284*, 34–43. [[CrossRef](#)]

116. Abidi, S.S.R. Healthcare Knowledge Management: The Art of the Possible. In *Knowledge Management for Health Care Procedures*; Riaño, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–20.
117. Ossowski, S. *Agreement Technologies*; Law, Governance and Technology Series; Springer: Amsterdam, The Netherlands, 2012; Volume 8, doi:10.1007/978-94-007-5583-3. [CrossRef]
118. Garcia, A.J.; Simari, G.R. Defeasible logic programming: An argumentative approach. *Theory Pract. Log. Program.* **2004**, *4*, 95–138. [CrossRef]
119. Modgil, S.; Prakken, H. The ASPIC+ framework for structured argumentation: A tutorial. *Argum. Comput.* **2014**, *5*, 31–62. [CrossRef]
120. Dung, P.M.; Kowalski, R.A.; Toni, F. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*; Springer: Amsterdam, The Netherlands, 2009; pp. 199–218.
121. Solarte-Vasquez, M.C.; Järv, N.; Nyman-Metcalf, K. Usability factors in transactional design and smart contracting. In *The Future of Law and eTechnologies*; Kerikmäe, T., Rull, A., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 149–176, doi:10.1007/978-3-319-26896-5\_8. [CrossRef]
122. Kerikmäe, T.; Hoffmann, T.; Chochia, A. Legal technology for law firms: Determining roadmaps for innovation. *Croat. Int. Relations Rev.* **2018**, *24*, 91–112. [CrossRef]
123. Prakken, H.; Sartor, G. Law and logic: A review from an argumentation perspective. *Artif. Intell.* **2015**, *227*, 214–245. [CrossRef]
124. Guangzuo, C.; Fei, C.; Hu, C.; Shufang, L. OntoEdu: A case study of ontology-based education grid system for e-learning. In Proceedings of the GCCCE Global Chinese Conference on Computers in Education, Hong Kong, China, 23–27 May 2004; pp. 1–9.
125. Sancho, P.; Martínez, I.; Fernández-Manjón, B. Semantic Web Technologies Applied to e-learning Personalization in <e-aula>. *J. Univers. Comput. Sci.* **2005**, *11*, 1470–1481.
126. Almohammadi, K.; Hagra, H. An adaptive fuzzy logic based system for improved knowledge delivery within intelligent E-Learning platforms. In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013), Hyderabad, India, 7–10 July 2013; pp. 1–8, doi:10.1109/FUZZ-IEEE.2013.6622350. [CrossRef]
127. Chrysafiadi, K.; Virvou, M. PeRSIVA: An empirical evaluation method of a student model of an intelligent e-learning environment for computer programming. *Comput. Educ.* **2013**, *68*, 322–333. [CrossRef]
128. Gong, T.; Cai, Z. An immune agent for web-based AI course. *Int. J. E-Learn.* **2006**, *5*, 493–506.
129. Yacef, K. The Logic-ITA in the classroom: A medium scale experiment. *Int. J. Artif. Intell. Educ.* **2005**, *15*, 41–62.
130. Pedro, F.; Subosa, M.; Rivas, A.; Valverde, P. *Artificial Intelligence in Education: Challenges and Opportunities for Sustainable Development*; Technical Report Number ED-2019/WS/8, January 2019; UNESCO: Paris, France, 2019. Available online: <https://unesdoc.unesco.org/ark:/48223/pf0000366994> (accessed on 21 March 2020).
131. Tuomi, I. *The Impact of Artificial Intelligence on Learning, Teaching, and Education*; Technical Report, JRC Working Paper; Joint Research Centre (Seville site): Seville, Spain, 2018.
132. Baheti, R.; Gill, H. Cyber-physical systems. *Impact Control Technol.* **2011**, *12*, 161–166.
133. Akbarinia, R.; Martins, V.; Pacitti, E.; Valduriez, P. Design and implementation of Atlas P2P architecture. *Glob. Data Manag.* **2006**, *8*, 98.
134. Simonis, H. Building Industrial Applications with Constraint Programming. In *Constraints in Computational Logics, Proceedings of the Theory and Applications International Summer School (CCL '99), Gif-sur-Yvette, France, 5–8 September 1999*; Goos, G., Hartmanis, J., van Leeuwen, J., Comon, H., Marché, C.M., Treinen, R., Eds.; Revised Lecture; Springer: Cham, Switzerland, 2001; pp. 271–309, doi:10.1007/3-540-45406-3\_6. [CrossRef]
135. Simonis, H.; Charlier, P.; Kay, P. Constraint handling in an integrated transportation problem. *IEEE Intell. Syst. Their Appl.* **2000**, *15*, 26–32. [CrossRef]
136. Baues, G.; Kay, P.; Charlier, P. Constraint based resource allocation for airline crew management. *Proc. ATTIS* **1994**, *94*.
137. Wallace, M. Practical applications of constraint programming. *Constraints* **1996**, *1*, 139–168. [CrossRef]
138. Simonis, H. Application development with the CHIP system. In *Lecture Notes in Computer Science, Proceedings of the Constraint Databases and Applications, Friedrichshafen, Germany, 8–9 September 1996*; Kuper, G.W.M., Ed.; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1034, pp. 1–21.

139. Bistarelli, S. Soft Constraint Solving and Programming: A General Framework. Ph.D. Thesis, Computer Science Department, University of Pisa, Pisa, Italy, 1 March 2001.
140. Ferrein, A.; Lakemeyer, G. Logic-based robot control in highly dynamic domains. *Robot. Auton. Syst.* **2008**, *56*, 980–991. [[CrossRef](#)]
141. Lytras, M.D. *Knowledge Management Strategies: A Handbook of Applied Technologies: A Handbook of Applied Technologies*; IGI Global: Hershey, PA, USA, 2008; Volume 5.
142. Chilwal, B.; Mishra, P.K. A Survey of Fuzzy Logic Inference System and Other Computing Techniques for Agricultural Diseases. In Proceedings of the International Conference on Intelligent Computing and Smart Communication, New Tehri, Uttarakhand, India, 20–21 April 2019; Singh Tomar, G., Chaudhari, N.S., Barbosa, J.L.V., Aghwariya, M.K., Eds.; Springer: Cham, Switzerland, 2020; pp. 1–6.
143. Rastogi, A.; Arora, R.; Sharma, S. Leaf disease detection and grading using computer vision technology & fuzzy logic. In Proceedings of the 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 19–20 February 2015; pp. 500–505.
144. Patil, P.; Kulkarni, U.; Desai, B.L.; Benagi, V.I.; Naragund, V.B. Fuzzy logic based irrigation control system using wireless sensor network for precision agriculture. In Proceedings of the 3rd National Conference on Agro-Informatics and Precision Agriculture (AIPA), Hyderabad, India, 1–3 August, 2012; pp. 262–269.
145. Zhu, A.; Yang, S.X. A fuzzy logic approach to reactive navigation of behavior-based mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04), New Orleans, LA, USA, 26 April–1 May 2004; Volume 5, pp. 5045–5050.
146. Center, B.; Verma, B.P. Fuzzy logic for biological and agricultural systems. In *Artificial Intelligence for Biology and Agriculture*; Springer: Cham, Switzerland, 1998; Volume 12, pp. 213–225.
147. Tahmasebi, P.; Hezarkhani, A. A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. *Comput. Geosci.* **2012**, *42*, 18–27. [[CrossRef](#)]
148. Finucane, C.; Jing, G.; Kress-Gazit, H. LTLMoP: Experimenting with language, temporal logic and robot control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1988–1993.
149. Fainekos, G.E.; Girard, A.; Kress-Gazit, H.; Pappas, G.J. Temporal logic motion planning for dynamic robots. *Automatica* **2009**, *45*, 343–352. [[CrossRef](#)]
150. Kloetzer, M.; Belta, C. Temporal Logic Planning and Control of Robotic Swarms by Hierarchical Abstractions. *IEEE Trans. Robot.* **2007**, *23*, 320–330. [[CrossRef](#)]
151. Smolensky, P. Connectionist AI, symbolic AI, and the brain. *Artif. Intell. Rev.* **1987**, *1*, 95–109. [[CrossRef](#)]
152. Sun, R. Artificial Intelligence: Connectionist and Symbolic Approaches. In *International Encyclopedia of the Social and Behavioral Sciences*; Pergamon: Izmir, Turkey; Elsevier: Amsterdam, The Netherlands, 2001; pp. 783–789.
153. Hendler, J. Avoiding Another AI Winter. *IEEE Intell. Syst.* **2008**, *23*, 2–4. [[CrossRef](#)]
154. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson Education: London, UK, 2003.
155. Hoehndorf, R.; Queralt-Rosinach, N. Data science and symbolic AI: Synergies, challenges and opportunities. *Data Sci.* **2017**, *1*, 27–38. [[CrossRef](#)]
156. Calegari, R.; Ciatto, G.; Dellaluce, J.; Omicini, A. Interpretable Narrative Explanation for ML Predictors with LP: A Case Study for XAI. In Proceedings of the WOA 2019—20th Workshop “From Objects to Agents”, CEUR Workshop Proceedings, Parma, Italy, 26–28 June 2019.
157. Calegari, R.; Ciatto, G.; Mariani, S.; Denti, E.; Omicini, A. LPaaS as Micro-intelligence: Enhancing IoT with Symbolic Reasoning. *Big Data Cogn. Comput.* **2018**, *2*. [[CrossRef](#)]
158. Besold, T.R.; d’Avila Garcez, A.S.; Bader, S.; Bowman, H.; Domingos, P.M.; Hitzler, P.; Kühnberger, K.; Lamb, L.C.; Lowd, D.; Lima, P.M.V.; et al. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. *CoRR* **2017**, *abs/1711.03902*.
159. Medsker, L.R. *Hybrid Intelligent Systems*; Springer Science & Business Media: Cham, Switzerland, 2012.
160. Corchado, E.; Kurzynski, M.; Wozniak, M. In *Hybrid Artificial Intelligent Systems, Proceedings of the 6th International Conference, HAIS 2011, Wroclaw, Poland, 23–25 May 2011*; Springer: Cham, Switzerland, 2011; Volume 667.
161. Nauck, D.; Klawonn, F.; Kruse, R. *Foundations of Neuro-Fuzzy Systems*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1997.
162. Sahoo, S.; Panda, S.K.; Pratihari, D.K.; Mukhopadhyay, S. Prediction of Step Length Using Neuro-Fuzzy Approach Suitable for Prosthesis Control. *IEEE Trans. Instrum. Meas.* **2020**, *1*, 1. [[CrossRef](#)]



163. Sohani, M.; Makki, B.; Sadati, N.; Kermani, K.K.; Riazati, A. A Neuro-Fuzzy approach to diagnosis of neonatal jaundice. In Proceedings of the 1st Bio-Inspired Models of Network, Information and Computing Systems, Madonna di Campiglio, Italy, 11–13 December 2006; pp. 1–4.
164. Wu, Y.; Zhang, B.; Lu, J.; Du, L.; Du, K.L. Fuzzy Logic and Neuro-fuzzy Systems: A Systematic Introduction. *Int. J. Artif. Intell. Expert Syst.* **2011**, *2*, 2011–2047.
165. Garcez, A.S.D.; Lamb, L.C.; Gabbay, D.M. *Neural-Symbolic Cognitive Reasoning*; Springer Publishing Company, Inc.: Cham, Switzerland, 2008.
166. Garcez, A.D.; Gori, M.; Lamb, L.C.; Serafini, L.; Spranger, M.; Tran, S.N. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *J. Appl. Logics IfCoLoG J. Logics Their Appl.* **2019**, *6*, 611.
167. Makki, B.; Hosseini, M.N. Some refinements of the standard autoassociative neural network. *Neural Comput. Appl.* **2013**, *22*, 1461–1475. [[CrossRef](#)]
168. Jain, A.K.; Mao, J.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *Computer* **1996**, *29*, 31–44. [[CrossRef](#)]
169. Garcez, A.D.; Besold, T.R.; De Raedt, L.; Földiák, P.; Hitzler, P.; Icard, T.; Kühnberger, K.U.; Lamb, L.C.; Miikkulainen, R.; Silver, D.L. Neural-symbolic learning and reasoning: Contributions and challenges. In Proceedings of the AAAI Spring Symposium Series, Stanford University, Palo Alto, CA, USA, 23–25 March 2015.
170. Pinkas, G. Symmetric neural networks and propositional logic satisfiability. *Neural Comput.* **1991**, *3*, 282–291. [[CrossRef](#)]
171. Guidotti, R.; Monreale, A.; Turini, F.; Pedreschi, D.; Giannotti, F. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv. (CSUR)* **2019**, *51*, 1–42, doi:10.1145/3236009. [[CrossRef](#)]
172. Andrews, R.; Diederich, J.; Tickle, A.B. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowl.-Based Syst.* **1995**, *8*, 373–389. [[CrossRef](#)]
173. d’Avila Garcez, A.S.; Broda, K.; Gabbay, D.M. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artif. Intell.* **2001**, *125*, 155–207. [[CrossRef](#)]
174. Bologna, G.; Hayashi, Y. A Comparison Study on Rule Extraction from Neural Network Ensembles, Boosted Shallow Trees, and SVMs. *Appl. Comput. Intell. Soft Comput.* **2018**, *2018*, 20. [[CrossRef](#)]
175. Bologna, G.; Hayashi, Y. A rule extraction study on a neural network trained by deep learning. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2016), Vancouver, BC, Canada, 24–29 July 2016; pp. 668–675, doi:10.1109/IJCNN.2016.7727264. [[CrossRef](#)]
176. Frosst, N.; Hinton, G.E. Distilling a Neural Network Into a Soft Decision Tree. In Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 Co-Located with 16th International Conference of the Italian Association for Artificial Intelligence (AI\*IA 2017), CEUR Workshop Proceedings, Bari, Italy, 14–17 November 2017; Besold, T.R., Kutz, O., Eds.; 2017; Volume 2071.
177. Johansson, U.; Niklasson, L. Evolving decision trees using oracle guides. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009, Part of the IEEE Symposium Series on Computational Intelligence, Nashville, TN, USA, 30 March–2 April 2009; pp. 238–244, doi:10.1109/CIDM.2009.4938655. [[CrossRef](#)]
178. Krishnan, R.; Sivakumar, G.; Bhattacharya, P. Extracting decision trees from trained neural networks. *Pattern Recognit.* **1999**, *32*, 1999–2009. [[CrossRef](#)]
179. Hruschka, E.R.; Ebecken, N.F.F. Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. *Neurocomputing* **2006**, *70*, 384–397. [[CrossRef](#)]
180. Zhou, Z.H.; Jiang, Y.; Chen, S.F. Extracting symbolic rules from trained neural network ensembles. *AI Commun.* **2003**, *16*, 3–15.
181. Craven, M.W.; Shavlik, J.W. Extracting Tree-Structured Representations of Trained Networks. In Proceedings of the 8th International Conference on Neural Information Processing Systems (NIPS’95), Denver, CO, USA, 27–30 November 1995; Touretzky, D.S., Mozer, M., Hasselmo, M.E., Eds.; MIT Press: Cambridge, MA, USA, 1995; pp. 24–30.
182. Augasta, M.G.; Kathirvalavakumar, T. Reverse Engineering the Neural Networks for Rule Extraction in Classification Problems. *Neural Process. Lett.* **2012**, *35*, 131–150. [[CrossRef](#)]



183. Sato, M.; Tsukimoto, H. Rule extraction from neural networks via decision tree induction. In Proceedings of the IJCNN'01, International Joint Conference on Neural Networks, Proceedings (Cat. No.01CH37222), Washington, DC, USA, 15–19 July 2001; Volume 3, pp. 1870–1875, doi:10.1109/IJCNN.2001.938448. [\[CrossRef\]](#)
184. Kahramanli, H.; Allahverdi, N. Rule extraction from trained adaptive neural networks using artificial immune systems. *Expert Syst. Appl.* **2009**, *36*, 1513–1522. [\[CrossRef\]](#)
185. 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. *Inf. Fusion* **2020**, *58*, 82–115. [\[CrossRef\]](#)
186. Došilović, F.K.; Brčić, M.; Hlupić, N. Explainable artificial intelligence: A survey. In Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 210–215.
187. Muggleton, S. Inverse entailment and Progol. *New Gener. Comput.* **1995**, *13*, 245–286. [\[CrossRef\]](#)
188. Muggleton, S.; Buntine, W. Machine invention of first-order predicates by inverting resolution. In *Machine Learning Proceedings 1988*; Elsevier: Amsterdam, The Netherlands, 1988; pp. 339–352.
189. Muggleton, S.; Santos, J.; Tamaddoni-Nezhad, A. ProGolem: A system based on relative minimal generalisation. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Inductive Logic Programming, Leuven, Belgium, 2–4 July 2009*; De Raedt, L., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5989, pp. 131–148.
190. Bordini, R.H.; Hübner, J.F.; Wooldridge, M. *Programming Multi-Agent Systems in AgentSpeak Using Jason*; Wiley Series in Agent Technology; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2007, Volume 8, doi:10.1002/9780470061848. [\[CrossRef\]](#)
191. Mochales, R.; Moens, M.F. Argumentation mining. *Artif. Intell. Law* **2011**, *19*, 1–22. [\[CrossRef\]](#)
192. Denti, E.; Natali, A.; Omicini, A. Programmable Coordination Media. In *Coordination Languages and Models, Proceedings of the 2nd International Conference (COORDINATION'97), Berlin, Germany, 1–3 September 1997*; Garlan, D., Le Métayer, D., Eds.; LNCS; Springer: Berlin, Germany, 1997; Volume 1282, pp. 274–288, doi:10.1007/3-540-63383-9\_86. [\[CrossRef\]](#)
193. De Angelis, F.L.; Di Marzo Serugendo, G. Logic Fragments: Coordinating Entities with Logic Programs. In *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques*; Margaria, T., Steffen, B., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 589–604.
194. Gardelli, L.; Viroli, M.; Omicini, A. Combining Simulation and Formal Tools for Developing Self-Organizing MAS. In *Multi-Agent Systems: Simulation and Applications*; Uhrmacher, A.M., Weyns, D., Eds.; Computational Analysis, Synthesis, and Design of Dynamic Systems; CRC Press: Boca Raton, FL, USA, 2009; Chapter 5, pp. 133–165.
195. Calegari, R.; Denti, E.; Mariani, S.; Omicini, A. Logic Programming as a Service. *Theory Pract. Log. Program.* **2018**, *18*, 846–873. doi:10.1017/S1471068418000364. [\[CrossRef\]](#)
196. Goranko, V.; Montanari, A.; Sciavicco, G. A road map of interval temporal logics and duration calculi. *J. Appl. Non-Class. Logics* **2004**, *14*, 9–54. [\[CrossRef\]](#)
197. Hölldobler, S.; Lutz, C.; Wansing, H. *Logics in Artificial Intelligence*; Springer: Cham, Switzerland, 2008; Volume 5293.
198. Della Monica, D.; Goranko, V.; Montanari, A.; Sciavicco, G. Interval temporal logics: A journey. *Bull. EATCS* **2013**, *3*, 105.
199. Larus, J.R.; Rajwar, R. Transactional memory. *Synth. Lect. Comput. Archit.* **2007**, *1*, 1–226.
200. Jacobsen, D.A.; Senocak, I. Multi-level parallelism for incompressible flow computations on GPU clusters. *Parallel Comput.* **2013**, *39*, 1–20. [\[CrossRef\]](#)

