LOGICS FOR ARTIFICIAL INTELLIGENCE

LUIGIA CARLUCCI AIELLO
Dipartimento di Informatica e Sistemistica "Antonio Ruberti" - Università degli Studi di Roma "La Sapienza"

ENRICO GIUNCHIGLIA
Dipartimento di Informatica Sistemistica e Telematica - Università degli Studi di Genova

LUCIANO SERAFINI
ITC-irst - Trento

1 Introduction

Logical languages were developed for humans, with the main objective of stating what is a *correct argument*. Artificial Intelligence (AI) provided a new motivation and application domain for logicians.

The main demand of AI to logics is to provide a formal language, a semantics, and a set of proof procedures that allow us to represent the various aspects of knowledge, and to reason about it. AI artifacts, such as robots, agents, "intelligent" programs, etc. act in a (sometimes virtual) world, and take decisions on the basis of what they "know" about this world. Here the need arises to be able to represent and use knowledge. Hence, speaking of Logics for AI largely means speaking of "Knowledge Representation and Reasoning."

Speaking of Logics for AI, one cannot avoid recalling the seminal work of John McCarthy. In particular, in the paper "Generality in Artificial Intelligence" [22], he explains the need of using logic to represent knowledge in a computer as a way to write general programs, i.e., programs which are able to work and adapt in a large number of situations.

> *. . . The 1958 idea for increasing generality [of programs] was to use logic to express facts in a way independent of the way the facts might subsequently be used. It seemed then and still seems that humans communicate mainly in declarative sentences rather than in programming languages for good objective reasons that will apply whether the communicator is a human, a creature from Alpha Centauri or a computer program. Moreover, the advantages of declarative information also apply to internal representation. The advantage of declarative information is one of generality. The fact that when two objects collide they make a noise may be used in particular situations to make a noise, to avoid making noise, to explain a noise or to explain the absence of noise. [22]*

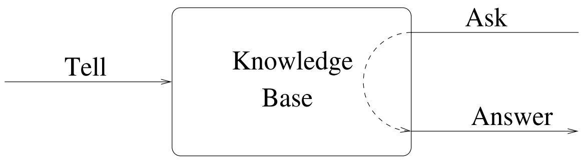

Figure 1: The structure of a logic-based system

So, the focus moved from "describing human knowledge (truth) on paper" to "realizing useful knowledge in a computer."

Why a logic language? It is declarative, it has a formal semantics, it is endowed with a reasoning apparatus, it is high level (i.e., independent of machine implementations), and it is understandable by humans. The structure of a logic-based system for representing and reasoning about knowledge is summarized in the picture in Figure 1 where, using the now standard nomenclature introduced by Hector Levesque in [20], "Tell" incorporates the new knowledge encoded in a statement, and "Ask" allows us to query what is known. Indeed, we expect that the "Answer" generated as response to an "Ask" follows from the state of the knowledge base as resulting from all the "Tell" actions.

Considering the figure, our knowledge base consists of a finite collection of formulas in a logical language. The main task of the knowledge base is to answer queries which are submitted to it, and to be able to incorporate new knowledge.

The following three sections are respectively devoted to the three aspects depicted in the figure, i.e., how knowledge can be represented in logic, how it is possible to answer queries, and how it is possible to incorporate new knowledge, possibly contradicting the existing one. Of course, it is not possible to accurately describe more than 50 years of research in logic and in logic in AI in a few pages. Our goal is thus to provide an introduction to the key issues arising when constructing a logic-based system, pointing to other articles of this collection and to the literature for details and applications.

For a somehow different treatment of the subject see [28], or the more classical books [19, 8, 18].

2 Logic-based Knowledge Bases

Logic has been invented with the main intent to formalize knowledge and correct reasoning, and has been widely used for this task, even if sometimes in an implicit way. For instance, standard relational data bases are based on logic [25], and many other complex structures such as frames [24] can be considered as syntactic variants of parts of first order logic [17].

It does not come with surprise that logic played a major role for formalizing knowledge in AI since the beginning. Indeed, researches in AI first adopted the existing logics developed by mathematicians, logicians and philosophers, such as classical (propositional and first order) logic, and modal logic to represent and reason about knowledge. However, the need to represent and efficiently reason about the many aspects of the world caused the birth of new logics. For instance, *Horn clause logics*, which gave rise to Prolog and Logic Programming (see the article on Automated Reasoning in this collection), or *description logics* (see the article on Ontologies and Description Logics in this collection), which are at the basis of current systems for representing ontologies and taxonomic knowledge. Similarly, the need to formalize actions and their effects on the state of the world stimulated the development of two main families of logics, *action languages* [10, 11] and the *Situation Calculus* [23, 27]. Action logics can also be seen as *spatial* and *temporal logics* (see, e.g., [1] and [9]), i.e., as logics formalizing the two dimensions -space and time- in which every entity (be it artificial or not) lives and moves. In addition, in order to take into account the fact that sometimes sentences in the real world are not sharply true or false, many valued logics [15] have been introduced, and also fuzzy logics (see the seminal paper [30] and the more recent [16]), which allow sentences to assume a truth value that goes from 0 (false) to 1 (true) in a continuum.

As the above few examples make clear, a wide range of logics have been developed. Grouping them on the basis of the specific aspect of the knowledge they are meant to represent would end up in a quite long list. Further, it would not provide a taxonomy of the existing logics, since many of them have been developed to formalize more than one aspect, e.g., both space and time.

It makes more sense to classify existing logics according to the features of the language being used. This is the guideline for the rest of the section.

Classical logics Classical logics comprehend propositional, first order, second order and - more generally - higher order logics. Going up in the hierarchy the expressive power increases. Propositional languages are the simplest ones; they are based on a set of atomic statements, which are meant to represent *propositions*, namely facts that can be true/false in a given world.

Atomic sentences can be composed via truth functional connectives, such as conjunction, disjunction, negation, implication etc, to form complex statements.

The main limitation in the expressiveness of the propositional language is its lack of generality. For example, in order to express a general fact such as - for instance - “every block is on the table,” one has to state this property for each single block in the world.

Predicate logics overcome the limitations of propositional languages by allowing a better granularity in the construction of atomic statements. Namely, the language is extended in order to include *terms*, where a term denotes an object in the world, and *relational symbols of arity* n ($n \geq 0$), representing a relation between n objects. Within predicate languages it is possible to construct an atomic statement by applying a relational symbol of arity n to a n-tuple of terms, and a propositional atomic statement corresponds to a relational symbol of arity 0. Further, in the first order case, the language allows for *variables* ranging over objects. Existential (resp. universal) quantification over the variable x is denoted with $\exists x$ (resp. $\forall x$) and allows us to represent statement like “there exists an object x such that...” and “for each object x ....” In second order languages, variables can range over sets of tuples of objects (i.e., relations) and this indeed gives more expressive power. Among the second order logics, a special role is played by Quantified Boolean Logic (QBL). Technically speaking, QBL is a second order logic, but all the predicates and variables ranging over relations have arity 0: essentially, it is propositional logic enhanced with quantifiers having variables ranging over the truth values true and false. Classical QBL is not more expressive than classical propositional logic, but it is exponentially more succinct [3].

Modal logics The use of classical logic to represent knowledge is based on the underlying hypothesis that there is one unique world (or state of affairs) in which propositions can take some truth value. On the other hand, it is often the case that our knowledge is about other "possible worlds."

More formally, in classical logic the truth value of a complex formula depends on the truth of its components in the actual (unique) world. For instance, the truth value of $A \rightarrow B$ is a function of the truth value of A and that of B. There are cases, however, in which it is necessary to express knowledge on a formula without explicitly referring to its truth values in the actual/current world. For instance, the truth of the formula expressed by the natural language sentence

> 'If John inserts 1 cent in the slot machine then it's possible that John wins 10000\$' (1)

in the current situation is completely independent of the truth of the formulas expressed by the subsentences (1.1) 'John inserts 1 cents in the slot machine' and (1.2) 'John wins 10000\$'. Conversely, the truth of (1) depends on the truth of (1.1) and (1.2) in *all the possible situations* which are reachable from a situation in which John inserts 1 cent in the slot machine. Thus, a sentence like (1) cannot be expressed following the principles outlined for classical logic. Instead, it is possible to introduce additional symbols, called *modalities*, which, applied to a proposition ϕ, express the fact that the truth value of the resulting proposition depends (also) on the values of ϕ in a set of worlds considered reachable from the current one. Typical examples of modalities which have been introduced in philosophy, mathematics and AI are those representing knowledge, belief, desire, and intention, to cite just a few. Indeed AI fostered the development of a very large number of modal languages, each differing from the other for the properties of the modalities used. In Table 1 we summarize the most important modal logics, showing an example of the formulas with the corresponding intuitive meaning. The relative facility with which these modalities can be introduced and semantically defined makes modal logics suited for the formalization of many knowledge representation tasks, such as knowledge and time (see, e.g., [7, 6, 14]). The semantics of a modal logic is usually given in terms of a *Kripke structure*, defined as a triple $\langle W, R, S\rangle$ where W is a set (called the set of worlds), $R \subseteq W \times W$ defines the accessibility relation among worlds (intuitively, $\langle w_1, w_2\rangle \in R$ means that the truth of modal sentences in w_1 depends on what is true in w_2), and S specifies which atomic proposition is true in each world $w \in W$.

Meta logics An alternative approach to solve the problem of expressing properties of formulas which are not functionally definable from their truth value,

Alethic modalities	
$\Box\phi$	ϕ is necessarily true
$\Diamond\phi$	ϕ is possibly true (equivalent to $\neg\Box\neg\phi$, i.e., ϕ is not necessarily false)
Epistemic modalities	
$\mathsf{K}_a\phi$	Agent a knows that ϕ is true
$\mathsf{B}_a\phi$	Agent a believes that ϕ is true
Linear Temporal modalities	
$\mathsf{X}\phi$	ϕ will be true at the next state of the world
$\mathsf{G}\phi$	ϕ will be always true
$\mathsf{F}\phi$	ϕ will be eventually true (equivalent to $\neg\mathsf{G}\neg\phi$, i.e., ϕ will not be always false)
$\phi\mathsf{U}\psi$	ϕ is true until ψ eventually becomes true
Dynamic modalities	
$[\alpha]\,\phi$	after executing the program α, ϕ must be true
$\langle\alpha\rangle\phi$	after executing the program α, ϕ may be true
$[\alpha+\beta]\,\phi$	after the execution of either α or β, ϕ is must be true
$[\alpha;\beta]\,\phi$	after the execution of α followed by β, ϕ is must be true
$[\alpha^*]\,\phi$	after n executions of α, ϕ must be true
$[\psi?]\,\phi$	after testing if ψ is true, ϕ must be true
BDI-agents modalities	
$\mathsf{B}_a\phi$	agent a believes that ϕ is true
$\mathsf{D}_a\phi$	a desires (has the goal) to achieve ϕ
$\mathsf{I}_a\phi$	a has the current intention to achieve the goal ϕ
Deontic modalities	
$\mathsf{OB}\phi$	it is obligatory that ϕ
$\mathsf{PE}\phi$	it is permissible that ϕ (equivalent to $\neg\mathsf{OB}\neg\phi$, i.e., it is not obligatory that not ϕ)
$\mathsf{IM}\phi$	it is impermissible that ϕ (equivalent to $\mathsf{OB}\neg\phi$, i.e., it is obligatory that not ϕ)
$\mathsf{GR}\phi$	it is gratuitous that ϕ (equivalent to $\neg\mathsf{OB}\phi$, i.e., it is not obligatory that ϕ)
$\mathsf{OP}\phi$	it is optional that ϕ (equivalent to $\neg\mathsf{OB}\phi \wedge \neg\mathsf{OB}\neg\phi$, i.e., neither ϕ nor $\neg\phi$ are obligatory)
Spatial/topological modalities	
$\Box\phi$	ϕ is true in all the neighboring points
$\Diamond\phi$	ϕ is true in at least one neighboring point, (equivalent to $\neg\Box\neg\phi$, i.e., ϕ is not false in all the neighboring points)

Table 1: Modalities and their intuitive meaning

is obtained by allowing the language to speak about the sentences themselves. This idea was extensively used by logicians like Tarski, Gödel, etc., with the main objective of providing a formal definition of when a *statement is true*. Their original idea was to extend a first order language with *terms* which are intended to denote formulas. For instance, the constant $\lceil$*Maria_is_married_to_Giuseppe*$\rceil$ is intended to denote the atomic statement *Maria_is_married_to_Giuseppe*, of which it is a *reification*. Reification is similar to the use of quotation in speech reports in natural language; it allows us to speak about the reified formula, and the resulting language is a standard predicative one.

Arguably, the most famous language with reification is the Situation Calculus (first introduced by John McCarthy in the sixties [21, 23], and then further developed after the rejuvenating work by Ray Reiter in the early nineties [26, 27]). Intuitively speaking, a situation corresponds to a snapshot of the world at some instant and can be represented via a term of sort *situation*. Thus, the language of the Situation Calculus is a predicate language which allows us to express properties of the states of the world.

In the Situation Calculus, reified statements are used to describe the states of the world and the effects of actions in a given state. However, the use of reification is not limited to this. From a logical point of view, reification allows one language to speak about another language. A language which is capable to describe another language is called *meta language* or meta (level) language, while the described language is called *object language*. Analogously, it is common to speak about meta knowledge and object knowledge in order to distinguish between facts expressed in the meta and object languages, respectively. Possibly, one of the first and most prominent proposals allowing for both meta and object level languages (separated from one another) is by Richard Weyhrauch [29], and was implemented in LISP in the FOL system. Since then there have been many uses of meta and object languages, see, e.g., [4], and [5] for a recent survey of the use of metalogic in knowledge representation.

Contextual logics All the previous approaches, with the exception of the works dealing with a meta-level theory distinct from an object-level theory, assume that the world is formalized in a unique logical system, i.e., in a unique logic. Contextual logics do not have this assumption. Any given world, problem, etc. is formalized via a set of theories, each with its own language, and meant to capture an aspect/piece of the whole scenario. The different theories (called *contexts*) communicate via a proper set of rules allowing us to derive one fact in one context on the basis of what has been derived in other contexts.

The study of a formal notion of context has a long history in various areas of AI. Again, the first reference can be traced back to Richard Weyhrauch and his work on mechanizing logical theories in FOL [29]. However, it became a widely discussed issue only in the late 1980s, when John McCarthy proposed the formalization of context as a crucial step toward the solution of the problem of generality [22], by claiming that no formal theory of common sense can get by without some formalization of context, as the representation of any piece of knowledge seems to crucially depend on the context in which it is asserted.

Along the same lines, though with a different emphasis, Fausto Giunchiglia [12] proposed the use of contexts in order to solve the *problem of locality*, namely the problem of modeling reasoning which uses only a subset of what is known about the world. The idea is that, while solving a problem, people do not use all their knowledge, but construct a "context" and use it *as if* it contained all relevant facts about the problem at hand. See [12, 13] for more details.

McCarthy and Giunchiglia provide different formalizations of contextual logics, while sharing the intuition that reasoning happens in contexts, and it is possible to switch from a context to another one, for example when a context is not adequate to solve a problem.

3 Querying a Knowledge Base

So far we described the most common logics used in AI, highlighting their expressive power. This influences the type of knowledge that can be expressed, hence represented, in a knowledge base. Referring again to Figure 1, we want also our knowledge base to be able to answer the queries submitted to it.

A knowledge base is a specification of a set of interpretations, where an *interpretation* I is a mathematical construction in which every formula ϕ in (the language of) the knowledge base can be evaluated to true or false. As a simple example, in classical propositional logic, an interpretation is a function mapping each atom to true or false, and evaluating a proposition ϕ corresponds to computing the truth value of ϕ according to the standard truth tables of the propositional connectives. In other logics, such as predicate and modal logics, interpretations are "richer" mathematical structures, hence evaluating the truth value of formulas is more difficult. Independently of the logic, an interpretation I is a *model* of a formula ϕ if the value of ϕ in I is *true*. The notion of model extends from a formula to a finite set of formulas, namely a knowledge base Γ. Let us call M_Γ the set of all models of Γ.

The notion at the basis of many reasoning tasks is that of *logical consequence*, which formalizes correct reasoning with a knowledge base: a formula ϕ is a logical consequence of a set of formulas Γ (written as $\Gamma \models \phi$) if every model in M_Γ is a model of ϕ. As a special case we get the notion of *validity* when Γ is empty (written as $\models \phi$).

Note that, it is standard to assume that a model of a set of formulas is a model of each formula in the set. This inevitably leads to *monotonic logics*: If we have two sets Γ and Γ' of formulas, if $\Gamma \subseteq \Gamma'$ then $M_{\Gamma'} \subseteq M_\Gamma$ and thus the set of logical consequences of Γ is a subset of the set of logical consequences of Γ'. In other words, adding a new piece of knowledge to the base cannot reduce the number of its consequences. In AI, it has been recognized that the logic behind many reasoning tasks is intrinsically nonmonotonic, as the logical consequences of a knowledge base can become false when new knowledge is acquired.

The formalization of nonmonotonic logics and reasoning has a long tradition in AI, starting from the famous Volume 13 of the "Artificial Intelligence Journal," published in 1980, and very well described (together with other approaches to nonmonotonic reasoning) in [8].

No matter whether the logic is monotonic or not, the question is how to effectively determine the facts which are true in each model of the knowledge base. The first observation is that we can expect this task to be more or less difficult according to the expressiveness of the logic. In general, the more expressive the logic, the more difficult it is to reason in it. For instance, checking whether a formula ϕ is a logical consequence of a finite set Γ of formulas is: (•) decidable and co-NP complete in classical propositional logic, (•) decidable and PSPACE complete for classical quantified boolean logic, (•) decidable but with varying complexity depending on the set of operators used for most propositional modal logics, (•) not decidable in classical first order logic. For a more in depth discussion of the various methods for checking whether $\Gamma \models \phi$ in various logics, we refer to the article on Automated reasoning in this collection. Here we briefly outline the methods based on deductive systems, and those based on specialized procedures.

Deductive systems The problem of defining the notion of logical consequence is one of the first problems that one has to tackle in proposing a new logic. As seen above, the standard way to define the notion of logical consequence is to define the notions of interpretation of the language and model of formulas.

The notion of logical consequences of a given set of formulas is a "semantic" one; alternatively we may proceed "syntactically" and introduce a notion of *deduction* or *derivation*. To this end, we provide a finite and possibly empty set of (schemas of) formulas, called *axioms*, and a finite set of *inference rules*, which allow us to infer/compute formulas, starting from the axioms. Axioms and inference rules define a *deductive system.*

For instance, the deductive system for the classical propositional logic proposed by Hilbert, consists of three axiom schemas,

$$\begin{array}{l} A \to (B \to A) \\ ((A \to (B \to C)) \to ((A \to B) \to (A \to C)) \\ (\neg B \to \neg A) \to ((\neg B \to A) \to B) \end{array}$$

and one inference rule

$$\frac{A, \quad A \to B}{B} \qquad \text{Modus Ponens}$$

Similarly, an axiomatization for the simple modal logics of knowledge called K consists of the above axioms and rules, extended with the axiom schema

$$\Box(A \to B) \to (\Box A \to \Box B)$$

and the inference rule

$$\frac{A}{\Box A} \qquad \text{Necessitation}$$

Axioms and inference rules are supposed to compute consequences in the logic independently of the specific content of the knowledge base. Given a knowledge base consisting of a set Γ of formulas, a formula ϕ is a *consequence* of Γ if there exists a *deduction of ϕ from* Γ, i.e., a sequence of formulas ending with ϕ such that each formula in the sequence is a formula in Γ, or is an axiom, or else is obtained by previous formulas in the sequence via the application of an inference rule.

In order for a deductive system to be of some interest in inferring logical consequences, it is of paramount importance that it is *correct* and *complete*, i.e., that it only derives formulas that are logical consequences of the initial set, and that all the logical consequences have a derivation using the deductive system, respectively.

Note that the above definition of derivation is again monotonic: adding a new fact to Γ enlarges the set of admissible deductions and thus the set of consequences.

Automated decision procedures The fact that the complexity of checking whether $\Gamma \models \phi$ is already relatively high in classical propositional logic, fostered the development of specialized procedures. The article on Automated Reasoning in this collection is dedicated to the state of the art in this area; here we only provide a sketchy description.

Most proof procedures for automated reasoning reduce the problem of logical consequence to a satisfiability test of a finite set of formulas Γ, denoted with SAT(Γ). SAT(Γ) checks whether there is a model in which all the formulas in Γ are true.

These procedures can be used also to check if a formula ϕ is a logical consequence of a finite set of basic facts Γ. Indeed SAT($\Gamma \cup \{\neg\phi\}$) returns false if and only if $\Gamma \models \phi$

Several procedures for un-satisfiability testing are available for various monotonic and nonmonotonic logics; the interested reader is referred to the article on Automated Reasoning.

4 Updating and Revising a Knowledge Base

An information system characterizes a view of the world with which it interacts; its input can take two forms; a query or an impetus for change. Physically, the information held by an information system might be a diagram, a graph, a spreadsheet, a database, a rulebase, or a more sophisticated formal entity. More often than not, information is uncertain and subject to change; this is the case even for simple database systems. Consequently, an information system requires a mechanism for modifying its view as more information about the world is acquired. The area of belief revision focus on modeling the behavior of an information system as it receives new information. In the literature, the operation of "updating" is usually distinguished from that of "revising" a knowledge base, though both operations are about inserting new information into it and both assume that the result should preserve as much information as possible.

Intuitively speaking, *updating* a knowledge base amounts to changing the information in order to account for some information about a new state of affairs. *Revising* a knowledge base amounts to changing the information in order to account for more information about the same state of affairs.

What makes belief revision non-trivial is that several ways for performing this operation are possible. For example, if the current knowledge includes the three facts A, B and $A \wedge B \rightarrow C$, the introduction of the new information $\neg C$, requires a revision of the knowledge base that can be done preserving consistency only by removing at least one of the three facts. In this case, there are at least three different ways for performing the revision, each corresponding to removing one of the three initial facts.

In the literature, the properties which should be satisfied by the operation of revision of a knowledge base at the light of new information have been extensively studied and discussed, starting from the seminal work of Alchourrón, Gärdenfors, and Makinson [2] which defined the postulates that should be satisfied by any operation of revision of a knowledge base.

5 Relation with other research areas

Research in logic interacts with almost any other research area in AI, and each of them provides an application domain for logic. Here we just list some, from the most prominent evergreen ones to the most fashionable at present: Planning, Machine Learning, Diagnosis, Natural Language Processing, Description Logics, Multiagent Systems, Question Answering, Expert Systems, Text Understanding, Knowledge Extraction, Semantic Integration, Knowledge Management. Some of these areas have a dedicated article in this collection and the interested reader is referred to them.

Research in logics for AI had also impact in areas of Computer Science that traditionally are not considered as part of AI, among them: Formal verification and specification of systems, Database and information integration, Distributed knowledge management, Semantic Web and Web Services.

Summing up, logic is for AI, and for Computer Science, what calculus is for control theory: it is thus not a surprise that it plays such an important role.

REFERENCES

[1] M. Aiello, I. Pratt-Hartman, and J. van Benthem, editors. *Handbook of Spatial Logics.* Kluwer–Springer, 2007. To appear.

[2] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the Logic of Theory Change: Partial Meet Contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[3] M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. Comparing Space Efficiency of Propositional Knowledge Representation Formalisms. In L. Carlucci Aiello, J. Doyle, and S. Shapiro, editors, *Procs of the Fifth Int. Conf. on K R & R*, pages 364–373, San Francisco, November 5–8 1996. Morgan Kaufmann.

[4] L. Carlucci Aiello and G. Levi. The Uses of Metaknowledge in AI Systems. In *Metalevel Architectures and Reflection*, pages 243–254. North-Holland, 1988. Appeared also in *ECAI 84: Advances in Artificial Intelligence*, North-Holland, 705–720, 1984.

[5] S. Costantini. Meta-Reasoning: A Survey. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, pages 253–288, London, UK, 2002. Springer-Verlag.

[6] A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *The Handbook of Theoretical Computer Science*, pages 997–1072. Elsevier Science Publishers, 1990.

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

[8] D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning.* Oxford University Press, 1994.

[9] A. Galton. Temporal Logic. In E.N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* `http://plato.stanford.edu/entries/`. 2003.

[10] M. Gelfond and V. Lifschitz. Representing Action and Change by Logic Programs. *Journal of Logic Programming*, 17:301–322, 1993.

[11] E. Giunchiglia and V. Lifschitz. An Action Language based on Causal Explanation: Preliminary Report. In *Proc. AAAI*, pages 623–630, 1998.

[12] F. Giunchiglia. Contextual Reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993.

[13] F. Giunchiglia and L. Serafini. Multilanguage Hierarchical Logics (or: How we can do without Modal Logics). *Artificial Intelligence*, 65:29–70, 1994. Also IRST-Technical Report 9110-07, IRST, Trento, Italy.

[14] R. Goldblatt. *Logics of Time and Computation. 2nd Ed.* CSLI Lecture Notes, 1992.

[15] S. Gottwald. Many-Valued Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* `http://plato.stanford.edu/entries/`. 2004.

[16] P. Hajek. Fuzzy Logic. In E.N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* `http://plato.stanford.edu/entries/`. 2002.

[17] P.J. Hayes. The Logic of Frames. In D. Metzing, editor, *Frame Conceptions and Text Understanding*, pages 46–61. de Gruyter, Berlin, 1980.

[18] C.J. Hogger, J.A. Robinson, A. Galton, and D.M. Gabbay, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4: Epistemic and Temporal Reasoning.* Oxford University Press, 1995.

[19] C.J. Hogger, J.A. Robinson, J.H. Siekmann, and D.M. Gabbay, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 1: Logical Foundations.* Oxford University Press, 1993.

[20] H. Levesque. Foundations of a Functional Approach to Knowledge Representation. *Artificial Intelligence*, 23(2):155–212, 1984.

[21] J. McCarthy. Situations, Actions and Causal Laws. Technical report, Stanford University, 1963. Reprinted in Semantic Information Processing (M. Minsky ed.), MIT Press, Cambridge, Mass., 1968, pp. 410-417.

[22] J. McCarthy. Generality in Artificial Intelligence. *Communications of ACM*, 30(12):1030–1035, 1987. Also in V. Lifschitz (ed.), *Formalizing common sense: papers by John McCarthy*, Ablex Publ., 1990, pp. 226–236.

[23] J. McCarthy and P. Hayes. Some Philosophical Problems from the standpoint of Artificial Intelligence. *Machine Intelligence*, 4:463–502, 1969.

[24] M. Minsky. A Framework for Representing Knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, New York, 1975.

[25] R. Reiter. Towards a Logical Reconstruction of Relational Database Theory. In M.L. Brodie, J. Mylopoulos, and J.W. Schmidt, editors, *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, pages 191–233. Springer-Verlag, 1984.

[26] R. Reiter. The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a Completeness Result for Goal Regression. In Vladimir Lifschitz, editor, *AI and MTC: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, San Diego, CA, 1991.

[27] R. Reiter. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems.* MIT Press, 2001.

[28] R. Thomason. Logic and Artificial Intelligence. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* `http://plato.stanford.edu/entries/`. 2005.

[29] R.W. Weyhrauch. Prolegomena to a Theory of Mechanized Formal Reasoning. *Artificial Intelligence*, 13(1):133–176, 1980.

[30] L. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.