Non-Classical Logics for Knowledge Representation and Reasoning

Laura Giordano [a], Valentina Gliozzi [b], Nicola Olivetti [c], Gian Luca Pozzato [b,*], and Camilla B. Schwind [d]

[a] *Dipartimento di Informatica - Universitá del Piemonte Orientale "Amedeo Avogadro", Viale Teresa Michel 11, 15121, Alessandria,*
Italy
E-mail: laura@mfn.unipmn.it

[b] *Dipartimento di Informatica - Universitá degli Studi di Torino, corso Svizzera 185, 10149, Torino,*
Italy
E-mail: {gliozzi,pozzato}@di.unito.it

[c] *LSIS - UMR CNRS 6168 Universitè Paul Cèzanne (Aix-Marseille 3), Campus de Saint Jérome, Avenue Escadrille Normandie-Niemen, 13397 Marseille Cedex 20*
France
E-mail: nicola.olivetti@lsis.org, nicola.olivetti@univ-cezanne.fr

[d] *Laboratoire d'informatique fondamentale, Parc Scientifique et Technologique de Luminy, 163 avenue de Luminy - case 901, 13288 Marseille Cedex 9*
France
E-mail: Camilla.Schwind@lif.univ-mrs.fr

Abstract. We briefly outline our research activity, started in the 90s, in the field of non-classical logics. In particular, we describe our activity on the use of non-classical logics for knowledge representation and on proof methods for non-monotonic and conditional logics.

Keywords: Non-classical logics, Knowledge Representation, Proof methods

1. Introduction

Our interest in the field of non-classical logic started with our work in Logic Programming at the beginning of the 90s. At that time we were working with Alberto on extensions of LP for dealing with hypothetical, conditional, defeasible and abductive reasoning. Those activities include the development of goal directed proof methods for Horn-like fragments of modal logics K, S4, S5 and their use in the definition of structuring constructs for logic programs; the study of negation as failure in a hypothetical logic programming (NProlog); the semantic characterization of truth maintenance systems (TMS), and its relation with the stable model semantics; the development of proof procedures for abductive logic programming; and the definition of a conditional logic programming language (CondLP). Since that time, we have started working on non-classical logics, focusing on the use of such logics in knowledge representation and on the development of proof methods for the automatization of conditional and non-monotonic logics.

Non-classical logics are widely used within the AI community, in the context of knowledge representation. In the following section, we describe the activity of our group in this area, concerning the use of

*Corresponding author. E-mail: pozzato@di.unito.it.

modal, temporal, conditional and non-monotonic logics for Reasoning about Actions and Change and for Belief Revision as well as in the specification and verification of multi-agent systems.

In section 3 we describe our activity regarding proof methods for non-classical logics and, in particular, for KLM non-monotonic logics and for Conditional Logics.

2. Knowledge Representation

As mentioned above, our activity in Knowledge Representation has been mainly concerned with the formalization of *change*, which is crucial both in the context of Reasoning about Actions as well as in the context of Belief Revision. Concerning Reasoning about Actions, we have proposed a few modal and temporal formalisms for modelling actions execution. In modal and temporal action theories, action execution is modelled by introducing action modalities, and the Ramification problem is addressed by making use of modal or temporal operators (see section 2.1). Such action theories have been used in the specification and verification of agent interaction protocols as well as in the specification, verification and composition of web services (section 2.2). Concerning Belief Revision, our research has mainly focused on the relationships between Belief Revision and Conditional Logics (section 2.3). In the following we describe the above activities, as well as our recent activity concerning reasoning about typicality and inheritance with exceptions in Description Logics (section 2.4).

2.1. Reasoning About Actions

The idea of representing actions as modalities comes from Dynamic Logics [18]. As observed in [20], classical dynamic logic adopts essentially the same ontology as McCarthy's situation calculus, by taking "the state of the world as primary, and encoding actions as transformations on states". Indeed, actions can be represented in a natural way by modalities, and states as sequences of modalities. In this setting, the action law, saying that action a has effect f when executed in a state in which P holds, can be expressed by the formula: $P \rightarrow [a]f$. Moreover, the precondition law, saying that action a is executable in a state in which condition C holds, can be expressed by the formula: $C \rightarrow < a > f$. Based on this idea, in [15] we have defined a modal action theory in which the frame problem is tackled by using a non-monotonic formalism which maximizes persistency assumptions and the ramification problem is tackled by introducing a modal causality operator which is used to represent causal dependencies among fluents. This action theory can also deal with incomplete initial states and with nondeterministic actions.

In [15], we have developed a temporal action theory based on a dynamic extension of Linear Temporal Logic (LTL). This logic, called DLTL (Dynamic Linear Time Temporal Logic) [19], extends LTL by strengthening the "until" operator by indexing it with regular programs. The advantage of using a linear time temporal logic is that it is a well established formalism for specifying the behavior of distributed systems, for which a rich theory has been developed and the verification task can be automated by making use of automata based techniques. In particular, for DLTL, in [13] a tableau-based algorithm for obtaining a Büchi automaton from a formula in DLTL has been presented, whose construction can be done on-the-fly, while checking for the emptiness of the automaton.

An alternative approach to reasoning about actions, based on Conditional Logics, has been proposed in [17].

2.2. Specification and Verification of Agent Interaction Protocols

The temporal action theory described above has been used in the specification and verification of communication protocols [16]. We have followed a social approach [25] to agent communication, where communication is described in terms of changes to the social relations between participants, and protocols in terms of creation, manipulation and satisfaction of commitments among agents. The description of the interaction protocol and of communicative actions is given in a temporal action theory, and agent programs, when known, can be specified as complex actions (regular programs in DLTL).

We have addressed several kinds of verification problems, including run-time verification of protocols as well as static verification of agent compliance with the protocols. Some of these problems can be formalized either as validity or as satisfiability problems in the temporal logic and can be solved by model checking techniques. Other problems, as compliance, are more challenging and require a special treatment [14]. The proposed approach has also been used in the specification of Web Services and, in particular, for reasoning about service composition [12].

2.3. Belief Revision

A lot of work has been devoted to the problem of finding a formal relation between Conditional Logics and Belief Revision [5,21]. Conditional Logics provide a semantics to conditional sentences of the form "if A, then B", denoted by $A \Rightarrow B$. Belief Revision is the area of Knowledge Representation that deals with the problem of how to integrate new information in a given belief set. The most known theory of Belief Revision is the so-called AGM theory (from Alchourrón, Gardenfors, and Makinson who first proposed it) that specifies a set of rationality postulates for integrating new information about a static domain into a belief set of the same domain.

The idea that there might be a relation between evaluation of conditional sentences and Belief Revision dates back to Ramsey, who proposed an acceptability criterion for conditionals in terms of belief change. According to this criterion, in order to decide whether to accept a conditional $A \Rightarrow B$ in a belief set K, one should add A to K by changing it as little as possible, and see if B follows. If it does, one should accept the conditional, otherwise one should reject it. In spite of the intuitiveness of Ramsey's criterion, its formalisation in the framework of Belief Revision is not straightforward. Many proposals, such as [5] run into the well-known Triviality Result, according to which there is no interesting Belief Revision system compatible with the proposed formalization. In [6,7] we have proposed a Conditional Logic that corresponds to Belief Revision, thus establishing a relation between the two domains, without running into the Triviality Result.

2.4. Reasoning About Typicality in Description Logics

The family of description logics (DLs) is one of the most important formalisms for knowledge representation. DLs correspond to tractable fragments of first order logic, and are reminiscent of the early semantic networks and of frame-based systems. They offer two key advantages: a well-defined semantics based on first-order logic and a good trade-off between expressivity and complexity. DLs have been successfully implemented by a range of systems and they are at the base of languages for the semantic web such as OWL.

A DL knowledge base comprises two components: (i) the TBox, containing the definition of concepts (and possibly roles), and a specification of inclusions relations among them, and (ii) the ABox containing instances of concepts and roles, in other words, properties and relations of individuals. Since the very objective of the TBox is to build a taxonomy of concepts, the need of representing prototypical properties and of reasoning about defeasible inheritance of such properties naturally arises. The traditional approach is to handle defeasible inheritance by integrating some kind of non-monotonic reasoning mechanism. This has led to study non-monotonic extensions of DLs. However, finding a suitable non-monotonic extension for inheritance reasoning with exceptions is far from obvious.

In [8], we have considered a novel approach to defeasible reasoning based on the use of a typicality operator $\mathbf{T}$. The intended meaning is that, for any concept C, $\mathbf{T}(C)$ singles out the instances of C that are considered as "typical" or "normal". Thus, an assertion as "normally students do not pay taxes" is represented by $\mathbf{T}(Student) \sqsubseteq \neg TaxPayer$. The DL obtained is called $\mathcal{ALC} + \mathbf{T}$.

In the logic $\mathcal{ALC} + \mathbf{T}$, one can have consistent knowledge bases containing the inclusions

$$\mathbf{T}(Student) \sqsubseteq \neg TaxPayer$$
$$\mathbf{T}(Student \sqcap Worker) \sqsubseteq TaxPayer$$
$$\mathbf{T}(Student \sqcap Worker \sqcap \exists HasChild.\top) \sqsubseteq \neg TaxPayer,$$

corresponding to the assertions: normally a student does not pay taxes, normally a working student pays taxes, but normally a working student having children does not pay taxes (because he is discharged by the government), etc.. Furthermore, if the ABox contains the information that for instance $\mathbf{T}(Student \sqcap Worker)(john)$, one can infer that $TaxPayer(john)$. In [10] we have developed a minimal model semantics for $\mathcal{ALC} + \mathbf{T}$ to maximize typical instances of a concept. By means of this semantics we are able to infer defeasible properties of (explicit or implicit) individuals. For instance, that allows us to make the inference above about $john$ also if we do not know explicitly that $\mathbf{T}(Student \sqcap Worker)(john)$, but we only know that $Student \sqcap Worker(john)$. In [10] a sound and complete tableau method for reasoning about such knowledge bases has been developed.

3. Proof Methods for Non-classical Logics

Our interest in the area of proof methods started with our work in Logic Programming.

At the beginning of the Nineties, our interest for proof methods for non-classical logics were mainly devoted to extend goal directed proof methods to non-classical logics, and, in particular to modal logics. In the same period, Dale Miller [22] was putting the basis of intuitionistic logic programming, based on the idea of having uniform proofs. Our work in this field was mainly concerned with modal extensions of logic programmimg [1,4] as well as with abductive, hypothetical and conditional extension of logic programming [3]. In the following, we describe our more recent activity concerning proof methods for non-monotonic and conditional logics.

3.1. Proof Methods for KLM Logics

In [9] we have introduced analytic tableau calculi for all non-monotonic logics introduced by Kraus, Lehmann, and Magidor (KLM). Such logics, namely **R**, **P**, **CL**, and **C**, have a preferential semantics in which a preference relation is defined among worlds or states. It has been observed that KLM logics correspond to the flat (i.e. unnested) fragment of well-known Conditional Logics.

Our tableau method provides a sort of run-time translation of **P** into modal logic G. The idea is simply to interpret the preference relation as an accessibility relation: a conditional $A \mid\sim B$ holds in a model if B is true in all minimal A-worlds, where a world w is an A-world if it satisfies A, and it is a minimal A-world if there is no A-world w' preferred to w. The relation with modal logic G is motivated by the fact that we assume, following KLM, the so-called *smoothness condition*, which ensures that minimal A-worlds exist whenever there are A-worlds, by preventing infinitely descending chains of worlds. This condition therefore corresponds to the finite-chain condition on the accessibility relation (as in modal logic G).

We have extended our approach to the cases of **CL** and **C** by using a second modality which takes care of states (intuitively, sets of worlds). Regarding **CL**, we have shown that we can map **CL**-models into **P**-models with an additional modality. In both cases, we can define a decision procedure to solve the validity problem in CoNP. Also, we have given a labelled calculus for the strongest logic **R**, where the preference relation is assumed to be modular. The calculus defines a systematic procedure which allows the satisfiability problem for **R** to be decided in nondeterministic polynomial time.

From the completeness of our calculi we get for free the finite model property for all the logics considered. With the exception of the calculus for **C**, in order to ensure termination, our tableau procedures for KLM logics do not need any loop-checking, nor blocking, nor caching machinery. Termination is ensured only by adopting a restriction on the order of application of the rules.

3.2. Proof Methods for Conditional Logics

In [24] we have introduced proof methods for some standard Conditional Logics. We have considered the *selection function* semantics. Intuitively, the selection function f selects, for a world w and a formula A, the set of worlds $f(w, A)$ which are "most similar to w" given the information A. In this respect, the selection function can be seen as a sort of modality indexed by formulas of the language. A conditional formula $A \Rightarrow B$ holds in a world w if B holds in all the worlds selected by f for w and A.

We have introduced cut-free sequent calculi for the basic Conditional Logic CK and for some of its extensions, namely CK+$\{$ID, MP, CS, CEM$\}$ including all the combinations of these extensions except those including *both* CEM and MP. Our calculi make use of labels representing possible worlds. Two types of formulas are involved in the rules of the calculi: world formulas of the form $x : A$, representing that A holds at world x, and transition formulas of the form $x \xrightarrow{A} y$, representing that $y \in f(x, A)$. The completeness of the calculi is an immediate consequence of the admissibility of cut.

We have also shown that one can derive a decision procedure from the cut-free calculi. Whereas the decidability of these systems was already proved by Nute (by a finite-model property argument), our calculi give the first *constructive* proof of decidability. As usual, the terminating proof search mechanism is obtained by controlling the backward application of some critical rules. By estimating the size of the finite derivations of a given sequent, we have also obtained a polynomial space complexity bound for the logics considered.

Our calculi can be the starting point to define goal-oriented proof procedures, according to the paradigm of Miller's Uniform Proofs recalled above. As a preliminary result, in [23] we have presented a goal-directed calculus for a fragment of CK and its extensions with MP and ID.

Proof methods for other Conditional Logics have been introduced in [11]. In detail, some modular labelled tableaux calculi have been defined for the Conditional Logic CE and its main extensions, including CV. These calculi are based on the preferential semantics of these logic formulated in terms of a ternary accessibility relation (or a family of binary accessibility relations). It is worth noticing that the flat fragments of CE and CV correspond, respectively, to KLM systems **P** and **R**. Once more, these calculi provide implementable decision procedures for the respective logics; their termination is obtained by imposing suitable blocking conditions. Moreover, given the semantic nature of tableaux, one obtains for free, by means of these calculi, a constructive proof of the finite model property of all conditional logics in exam.

4. Conclusions and Future Works

We believe that the temporal action theory we have described above can be profitably used for the verification of business process compliance. In this context, new issues arise such as, for instance, the problem of encoding norms which are inherently defeasible. Our first results on the topic have been presented in [2], where obligations in norms are captured by the notion of commitment, which is borrowed from the social approach to agent communication [25], and the compliance of a business process to norms is verified by making use of bounded model checking techniques.

Concerning reasoning about typicality in description logics, we are currently investigating the extension with the typicality operator of low complexity description logics such as $\mathcal{EL}$ and DL-Lite, with the aim of defining non-monotonic, but still tractable DLs.

Acknowledgements. This work has been partially supported the projects “ICT4LAW - ICT Converging on Law” and “MIUR PRIN 2008 - LoDeN”.

References

[1] M. Baldoni, L. Giordano, and A. Martelli, A modal extension of logic programming: Modularity, beliefs and hypothetical reasoning, *J. of Logic and Computation* **8(5)** (1998), 597–635.

[2] D. D’Aprile, L. Giordano, V. Gliozzi, A. Martelli, G.L. Pozzato, and D. Theseider Dupré, Verifying Business Process Compliance by Reasoning about Actions, in: *Proc. of CLIMA XI*, **LNAI 6245**, J. Dix, J. Leite, G. Governatori, and W. Jamroga, eds., Springer, 2010, pp. 99–116.

[3] D.M. Gabbay, L. Giordano, A. Martelli, N. Olivetti and M.L. Sapino, Conditional reasoning in Logic Programming, *J. of Logic Programming* **44(1-3)** (2000), 37–74.

[4] D.M. Gabbay and N. Olivetti, *Goal-Directed Proof Theory (Applied Logic Series V. 21)*, Springer, 2000.

[5] P. Gärdenfors, Belief Revisions and the Ramsey Test for Conditionals, *Philosoph. Review* **95** (1986), 81–93.

[6] L. Giordano, V. Gliozzi, and N. Olivetti, Iterated Belief Revision and Conditional Logic, *Studia Logica* **70(1)** (2002), 23–47.

[7] L. Giordano, V. Gliozzi, and N. Olivetti, Weak AGM postulates and strong Ramsey Test: A logical formalization, *Artificial Intelligence* **168(1-2)** (2005), 1–37.

[8] L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato, $\mathcal{ALC} + \mathbf{T}$: a Preferential Extension of Description Logics, *Fundamenta Informaticae* **96(3)** (2009), 341–372.

[9] L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato, Analytic Tableaux Calculi for KLM Logics of Nonmonotonic Reasoning, *ACM Trans. on Computational Logics (TOCL)* **10(3)** (2009).

[10] L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato, Reasoning About Typicality in Preferential Description Logics, in: *Proc. of JELIA 2008*, **LNAI 5293**, S. Hólldobler, C. Lutz, and H. Wansing, eds., Springer, 2008, pp. 192–205.

[11] L. Giordano, V. Gliozzi, N. Olivetti, and C.B. Schwind, Tableau Calculi for Preference-Based Conditional Logics, *ACM Transactions on Computational Logics (TOCL)* **10(3)** (2009).

[12] L. Giordano, A. Martelli, Web Service Composition in a Temporal Action Logic, *4th Int. Workshop on AI for Service Composition, AISC2006*, Riva del Garda, 28 August 2006.

[13] L. Giordano, A. Martelli, Tableau-Based Automata Construction for Dynamic Linear Time Temporal Logic, *Annals of Mathematics and Artificial Intelligence* **46(3)** (2006), 289-315.

[14] L. Giordano and A. Martelli, Verifying Agent Conformance with Protocols specified in a Temporal Action Logic, in: *Proc. of AI*IA 2007*, **LNAI 4733**, R. Basili and M.T. Pazienza, eds., Springer, Berlin, 2007, pp. 145–156.

[15] L. Giordano, A. Martelli and C. Schwind, Ramification and causality in a modal action logic, *Journal of Logic and Computation* **10(5)** (2000), 625–662.

[16] L. Giordano, A. Martelli, C. Schwind, Specifying and Verifying Interaction Protocols in a Temporal Action Logic, *J. of Applied Logic* **5(2)** (2007), 214-234.

[17] L. Giordano and C. Schwind, Conditional logic of actions and causation, *Artificial Intelligence* **157(1-2)** (2004), 239–279.

[18] D. Harel, First order dynamic logic, in *Extensions of Classical Logic, Handbook of Philosophical Logic II*, pp. 497–604, 1984.

[19] J.G. Henriksen and P.S. Thiagarajan, Dynamic Linear Time Temporal Logic, *Annals of Pure and Applied logic* **96(1-3)** (1999), 187–207.

[20] L.T. McCarty, Modalities over actions, I. model theory, in: *Proc. of KR ’94*, J. Doyle, E. Sandewall, P. Torasso, eds., Morgan Kaufmann, 1994, pp. 437–448.

[21] D. Makinson, The Gärdenfors impossibility theorem in nonmonotonic contexts, *Studia Logica* **49(1)** (1990), 1–6.

[22] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov, Uniform Proofs as a Foundation for Logic Programming, *Annals of Pure and Applied Logic* **51(1-2)** (1991), 125–157.

[23] N. Olivetti and G.L. Pozzato, Theorem Proving for Conditional Logics: CondLean and GOALD$\mathcal{U}$CK, *Journal of Applied Non-Classical Logics (JANCL)* **18(4)** (2008), 427–473.

[24] N. Olivetti, G.L. Pozzato, and C.B. Schwind, A Sequent Calculus and a Theorem Prover for Standard Conditional Logics, *ACM Transactions on Computational Logic (ToCL)* **8(4)** (2007).

[25] M. P. Singh, A Social Semantics for Agent Communication Languages, *Issues in Agent Communication 2000*, pp. 31-45.