Using Logic Programs to Model an Agent's Epistemic State

M. Capobianco C. I. Chesnevar

Grupo de Investigaci n en Inteligencia Artificial (GIIA)
Departamento de Ciencias de la Computaci n
Universidad Nacional del Sur
Bah a Blanca - Buenos Aires - ARGENTINA
e-mail: {mc,cic}@cs.uns.edu.ar

1 Introduction

The notion of *rational agency* was proposed by Russell [9] as an alternative characterization of *intelligence agency*. Loosely speaking, an agent is said to be rational if it performs the right actions according to the information it possesses and the goals it wants to achieve. Unfortunately, the enterprise of constructing a rational agent is a rather complex task. Although in the last few years there has been an intense flowering of interest in the subject, it is still in its early beginnings: several issues remain overlooked or addressed under too unrealistic assumptions.

As stated by Pollock [5], a rational agent should have models of itself and its surroundings, since it must be able to draw conclusions from this knowledge that compose its set of beliefs. Traditional approaches rely on multi-modal logics to represent the agent's epistemic state [7, 1]. Given the expressive power of these formalisms, their use yields proper theoretical models. Nevertheless, the advantages of these specifications tend to be lost in the transition towards practical systems: there is a tenuous relation between the implementations based on these logics and their theoretical foundations [8].

Modal logics systems suffer from a number of drawbacks, notably the well-known *logical omniscience problem* [10]. This problem arises as a by-product of the necessitation rule and the K axiom, present in any normal modal system. Together, these rules imply two unrealistic conditions: an agent using this system must know all the valid formulas, and its beliefs should be closed under logical consequence. These properties are overstrong for a resource-bounded reasoner to achieve them. Therefore, the traditional modal logic approach is not suitable for representing practical believers [11].

We intend to use logic programs as an alternative representation for the agent's epistemic state. This formalization avoids the aforementioned problems of modal logics, and admits a seamless transition between theory and practice. In the next section we detail our model and highlight its advantages. Next, section 3 presents some conclusions and reports on the forthcoming work.

2 Our proposal

Using logic programs to describe the agent's knowledge was previously addressed by Pereira *et. al.* in [4]. In this setting, an agent is defined as a logic program extended with explicit negation, and its beliefs are set by the well-founded semantics of this program. The weakness of this approach is that extended logic programming (ELP) does not constitute an appropriate formalism to accomplish this task. ELP can represent an interesting class of nonmonotonic knowledge bases in a rather simple way, but it cannot deal with partial and potentially contradictory information [3], an essential capability in practical agents. To solve this, a more expressive system is required.

Defeasible Logic Programming (DeLP) [2] is a newly developed formalism that combines the advantages of logic programming with *defeasible argumentation.* In DeLP an *argumentation system* is used for deciding between contradictory goals, through a *dialectical analysis.* Intuitively, an *argument* is a minimal set of rules supporting its conclusion. To answer a query q with respect to a certain defeasible logic program, arguments for and against q are formed in order to find a *justification*; that is, an argument that *defeats* all of its *counterarguments*. Since defeatears are also arguments, defeaters for the defeaters may exists. In consequence, it is necessary to define a dialectical analysis that establishes the status of an argument. Based on this analysis, a query q can obtain one of the following answers:

- **yes**: there exists a justified argument for q.
- **no**: no argument for q is ultimately accepted.
- **undecided**: there are arguments of equal strength for and against q.
- **unknown**: no argument can be built that supports q.

Describing an agent's epistemic state by means of DeLP is a promising solution, since it provides a good trade-off between expressivity and implementability. In our proposal, a *defeasible logic program* models the agent's knowledge, and the belief set is composed by the queries answered in a positive fashion with respect to this program.

The main advantage of this stance is that any agent based on DeLP inherit its ability to handle conflicting information. On the contrary, when using an extended logic program, additional mechanisms to enforce consistency must be specified. Particularly, in [4] contradictions are associated with integrity constraints that denote incompatible pieces of knowledge. If one of these rules is triggered, a revision procedure is started, whose objective is to find an stable model for the extended logic program by eliminating some of the conflicting beliefs. To do this, a preference relation between minimal non-contradictory models must be defined. However, designing suitable revision criteria is still an open problem.

Furthermore, there is a fundamental question regarding the essence of revision. The elimination of a certain belief withdraws information from the system. Suppose that a and b are conflicting beliefs, and the revision criterion chooses to remove b in order to restore consistency. Next, the system finds that a and a new belief c are in conflict, and decides to drop a from its knowledge base. With a defeated by c to continue in disbelieving b is unsound, but since it was previously removed it cannot be reinstated.

This situation is an unintended consequence of the revision process. We believe that any mechanism to enforce consistency entangles the framework, and may have unforeseen consequences (as shown by the previous example). These problems can be avoided using a formalism like DeLP, where there is no need to specify constraint rules or revision procedures.

3 Work in progress

We have outlined the intuitions supporting a promising formalization for the design of rational agents, that helps to minimize the existing gap between the formal theory and its practical implementation. This is only the beginning: some research is being conducted on defining an agent architecture according to the proposed ideas. This architecture must be able to accommodate all of the features required by a rational agent.

As remarked by Pollock [6], an agent able to interact in a complex dynamic environment cannot be provided with all the necessary information from its conception. In this situation, a mechanism to perceive and incorporate changes becomes imperative. As a result, we are extending our theory to supply the agent with perception capabilities.

Information attitudes are related to the knowledge an agent has about the world, whereas pro-attitudes are those that in some way guide the agents actions. As pointed out by Wooldridge and

Jennings [11], a rational agent must be represented in terms of at least one information attitude and one pro-attitude, since it will make choices and formulate intentions based on the information it possess. For this reason, both of them are needed to direct the agent's behavior. So far we have only considered beliefs and knowledge, without taking into account pro-attitudes, like *desires* and *intentions* [7, 11] whose role in the design of rational agents has been acknowledged in the philosophical and artificial intelligence literature. Consequently, our model has to be extended to reckon with this pro-attitudes.

References

[1] Cohen, P. R., and Levesque, H. J. Intention is choice with commitment. *Artificial Intelligence 42*, 3 (1990), 213{261.

[2] Garc a, A. J. Defeasible logic programming : definition and implementation. Master's thesis, Departamento de Ciencias de la Computacbn, Universidad Nacional del Sur, Bah a Blanca, Argentina, June 1997.

[3] Garc a, A. J., Simari, G. R., and Chesnevar, C. I. An Argumentative Framework for Reasoning with Inconsistent and Incomplete Information. In *Proceedings of the 13th European Conference on Artificial Intelligence, Workshop on Practical Reasoning and Rationality* (Aug. 1998), pp. 13{19. http://cs.uns.edu.ar/giia.html.

[4] Pereira, L. M., and Quaresma, P. Modelling Agent Interaction in Logic Programming. In *Proceedings of the 11th International Conference on Applications of Prolog* (Sept. 1998), O. Barenstein, Ed.

[5] Pollock, J. L. The Phylogeny of Rationality. *Computational Intelligence 17* (1993), 568{588.

[6] Pollock, J. L. Perceiving and Reasoning about a Changing World. *Computational Intelligence 14*, 4 (1998), 498{562.

[7] Rao, A. S., and Georgeff, M. P. Modeling Agents Within a BDI-Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning* (Apr. 1991), pp. 473{484.

[8] Rao, A. S., and Georgeff, M. P. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multi-Agent Systems* (San Francisco, CA, June 1995), V. Lesser, Ed., MIT Press, pp. 312{319.

[9] Russell, S. J. Rationality and Intelligence. *Artificial Intelligence 94*, 1{2 (1997), 57{77.

[10] Vardi, M. Y. On Epistemic Logic and Logical Omniscience. In *Proceedings of the First Conference on Theoretical Aspects of Reasoning About Knowledge* (1986), J. Y. Halpern, Ed., Morgan Kaufmann Publishers, pp. 293{305.

[11] Wooldridge, M., and Jennings, N. R. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review 10*, 2 (1995), 115{152.