International Journal of Approximate Reasoning 55 (2014) 739–761

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Semantics for possibilistic answer set programs: Uncertain rules versus rules with uncertain conclusions

Kim Bauters[a,∗,1], Steven Schockaert[b], Martine De Cock[a], Dirk Vermeir[c]

[a] Department of Applied Mathematics and Computer Science, Universiteit Gent, Krijgslaan 281 (S9), 9000 Gent, Belgium
[b] School of Computer Science & Informatics, Cardiff University, 5 The Parade, Cardiff CF24 3AA, United Kingdom
[c] Department of Computer Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium

ARTICLE INFO

Article history:
Received 18 January 2013
Received in revised form 8 September 2013
Accepted 8 September 2013
Available online 16 September 2013

Keywords:
Logic programming
Non-monotonic reasoning
Possibility theory
Uncertainty
Stable model semantics

ABSTRACT

Although Answer Set Programming (ASP) is a powerful framework for declarative problem solving, it cannot in an intuitive way handle situations in which some rules are uncertain, or in which it is more important to satisfy some constraints than others. Possibilistic ASP (PASP) is a natural extension of ASP in which certainty weights are associated with each rule. In this paper we contrast two different views on interpreting the weights attached to rules. Under the first view, weights reflect the certainty with which we can conclude the head of a rule when its body is satisfied. Under the second view, weights reflect the certainty that a given rule restricts the considered epistemic states of an agent in a valid way, i.e. it is the certainty that the rule itself is correct. The first view gives rise to a set of weighted answer sets, whereas the second view gives rise to a weighted set of classical answer sets.

1. Introduction

Answer set programming (ASP) is a form of logic programming which uses the stable model semantics to define negation-as-failure in a purely declarative way. An ASP program consists of a set of rules, e.g.

$$beach \leftarrow \neg weekday, not\ rain. \tag{1}$$

This rule intuitively encodes that we go to the beach when we know that today is not a weekday and when there is no indication of rain. Reasoning in ASP is non-monotonic because of the semantics of '*not*', which is called the negation-as-failure operator. In particular, '*not rain*' is true unless we have evidence that '*rain*' is true, e.g. we may read in the newspaper that rainy weather is predicted at the beach and we may consequentially need to revise our plans to go to the beach.

While ASP is well-suited for reasoning over incomplete information, it lacks the means to reason over uncertain information in a natural way. Nevertheless, uncertain information is an important and pervasive component of common-sense reasoning. For example, in (1) it may happen that we are driving an old car, in which case we may be uncertain as to whether or not we will reach the beach, even when all the premises are satisfied.

In this paper we are mainly interested in the potential of ASP as a tool for epistemic reasoning, i.e. for reasoning about what another agent believes. Under this view answer sets are interpreted as possible epistemic states of that agent. Rules are then seen as pieces of knowledge which constrain the epistemic state that an agent may have.

∗ Corresponding author. Tel.: +32 9 264 49 05.
E-mail addresses: kim.bauters@gmail.com (K. Bauters), s.schockaert@cs.cardiff.ac.uk (S. Schockaert), martine.decock@ugent.be (M. De Cock), dvermeir@vub.ac.be (D. Vermeir).

1 Funded by a joint Research Foundation-Flanders (FWO) project.

0888-613X/$ – see front matter
http://dx.doi.org/10.1016/j.ijar.2013.09.006

Answer sets, however, have a number of limitations for modeling epistemic states. One limitation is that an answer set can only express that the agent knows that a given literal is true, or that the agent has insufficient knowledge to assess the truth of the literal. An answer set cannot express in a natural way the belief of an agent that "*either literal* l_1 *or literal* l_2 *is true, but not both*". This issue has been the topic of [1,2]. An orthogonal issue is that classical ASP cannot intuitively be used to express that knowledge may be more or less certain. This particular issue is the focus of this paper.

There are two natural ways of introducing degrees of uncertainty in the setting of ASP. On the one hand, we may wish to model weighted epistemic states, in which an agent can be completely certain about the truth of some literals, while believing in the truth of other literals without complete certainty. On the other hand, we may wish to keep epistemic states Boolean, but rather express that the rules which constrain the possible epistemic states are not fully certain. In the latter case, an ASP program should correspond to a weighted set of classical answer sets. In this paper, we will compare and contrast both views, using possibility theory to model the semantics of both types of uncertainty degrees.

Combining ASP with possibility theory is not a new idea. In fact, two particular forms of possibilistic ASP were already investigated in [3] and [4]. Both approaches adhere to the first view, i.e. they model weighted epistemic states. The approaches do, however, differ in the way they treat negation-as-failure. In particular, the semantics from [3] consider a naf-literal of the form '*not* l' to be false as soon as there is *some* evidence that 'l' is true. The semantics from [4], on the other hand, adhere to the intuition that '*not* l' is certain to the extent that '$\neg l$' is possible. Which semantics to use is often dependent on the context of the problem. Consider the example:

$$\mathbf{1}: \neg breathing \leftarrow$$

$$\mathbf{1}: dead \leftarrow \neg breathing, \neg pulse$$

$$\mathbf{0.6}: dead \leftarrow \neg pulse$$

$$\mathbf{0.2}: dead \leftarrow \neg breathing$$

$$\mathbf{0.9}: first\_aid\_successful \leftarrow not\ dead$$

This example models the reasoning of the first responder arriving at the scene of an incident and faced with an unconscious victim. A quick examination shows that the victim is no longer breathing. From his experience, the first responder knows that lack of breathing does not necessarily indicate that the person is dead. However, if the victim also lacked a pulse, then this would considerably increase his certainty that the victim has died. The first responder has a strong certainty that applying first aid will be successful when there is no indication that the person is dead. Alternatively, we could replace the last rule with the rule '$\mathbf{0.9}$: *apply_first_aid* $\leftarrow$ *not dead*'. Then, the intuition of the last rule becomes that the first responder has a high certainty that he should at least attempt to apply first aid when there is no indication that the victim is dead.

Regardless of the last rule, both the semantics from [3] and [4] agree that $\neg$*breathing* is fully certain and *dead* is certain to degree 0.2. Given the original last rule, the semantics from [3] furthermore conclude that the certainty of *first_aid_successful* is 0, whereas the semantics from [4] conclude that *first_aid_successful* is certain to degree 0.8. The conclusion that we are certain of *first_aid_successful* to degree 0 is sound since *dead* is certain to degree 0.2 while the certainty of $\neg$*dead* is 0, i.e. we are more certain that the victim is dead than that he is alive. If, however, we consider the alternative last rule, then given the approach from [3] we conclude that the certainty of *apply_first_aid* is 0 and *apply_first_aid* is certain to degree 0.8 under the approach from [4]. In this case, however, we can argue that the conclusion must be that *apply_first_aid* is certain to degree 0.8 since we only have a low certainty that the victim is dead, i.e. we entertain a high possibility that the victim is still alive. As this example illustrates, the desired answer greatly depends on the particular understanding of the problem at hand. In Section 2.3 we recall the details of both approaches.

The rules in the example above are themselves not uncertain. Rather, the rules are considered valid, but they only allow us to reach conclusions with limited certainty. Often, however, we may be uncertain whether the rules are actually meaningful, e.g. when they are coming from possibly unreliable sources. Consider this example:

$$\mathbf{0.7}: paper\_title(title) \leftarrow$$

$$\mathbf{0.9}: author(John\_Doe) \leftarrow paper\_title(title)$$

$$\mathbf{0.2}: author(Jane\_Roe) \leftarrow paper\_title(title)$$

$$\mathbf{1}: \leftarrow author(John\_Doe), author(Jane\_Roe)$$

During a conference, a colleague shares the title of an interesting paper with us. We are quite certain that we recall the name of the title correctly and we would like to find out who the principal author of the paper is. We consult the university website, which in the past has given reliable answers. However, a quick search on the internet results in a different name for the same paper. Evidently, they cannot both be the principal author of the paper.

The uncertainty attached to each rule now expresses how certain we are that the information encoded in the rule is indeed valid. In particular, any world in which both John Doe and Jane Roe are the principal author of the paper can immediately be discarded due to the absolute certainty of the last rule. We do, however, acknowledge that neither of the candidate authors may be correct because we do not have absolute certainty as to whether we correctly recall the title