International Journal of Approximate Reasoning 64 (2015) 54–74

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Semantic-based construction of arguments: An answer set programming approach

Esteban Guerrero, Juan Carlos Nieves *, Helena Lindgren

Department of Computing Science, Umeå University, SE-901 87, Umeå, Sweden

ARTICLE INFO

Article history:
Received 15 May 2014
Received in revised form 22 June 2015
Accepted 29 June 2015
Available online 7 July 2015

Keywords:
Argumentation
Logic programming
Well-founded semantics
Argumentation tools
Stable model semantics
Answer set programming

ABSTRACT

In this paper, we introduce an argumentation approach which takes an extended logic program as input and gives a set of arguments with the respective disagreements among them as output. We establish the notion of an argument under the Well-Founded semantics and Stable semantics inferences, allowing us to identify arguments with stratified programs as support, even when the *input* for the argument engine is a non-stratified program. We propose a set of rationality postulates for argument-based systems under extended logic programs, which are based on a definition of closure for a set of clauses that considers the well-known Gelfond–Lifschitz reduction. We establish the conditions under which our approach satisfies these principles. In addition, we present a standalone argumentation-tool based on the XSB system which implements our argumentation approach.

1. Introduction

Common-sense reasoning is the type of reasoning that a person often performs when reasoning about what to do by evaluating the potential results of the different actions that (s)he can take [1]. Since the inception of this concept in the Artificial Intelligence field by authors such as McCarthy [2], common-sense reasoning has been a fundamental goal for different reasoning approaches like *non-monotonic reasoning* (NMR). NMR captures and represents *defeasible inference*, i.e., the kind of inference in everyday life in which conclusions are drawn tentatively, reserving the right to retract them in the light of further information [3]. Indeed, a number of "non-monotonic" logics [4–7] have been developed for capturing common-sense knowledge. In this setting, *Argumentation Theory* has emerged as a formalism for dealing with non-monotonic reasoning. Dung demonstrated in his seminal paper [8], that many of the major approaches to non-monotonic reasoning in Artificial Intelligence and Logic Programming (LP) are different forms of argumentation. In this scenario, different extensions of the so-called *abstract argumentation frameworks* have been developed [9–12], providing bases and analytic tools for non-monotonic reasoning, disregarding the structure or nature of the arguments themselves. An argument is the basic and fundamental element of these frameworks with a common structure *support-conclusion*. At the same time, different approaches [13–17] have instantiated these abstract frameworks into argument-based systems (ABS), building arguments, applying a given argumentation semantics and identifying accepted conclusions. In this setting, the underlying formalisms for knowledge representation play an important role. This is because the argument structure and its underlying formalism define conditions and restrictions with respect to the management of the knowledge bases; and furthermore, in the "quality" of the conclusions of such argument-based systems.

* Corresponding author.
E-mail addresses: esteban@cs.umu.se (E. Guerrero), jcnieves@cs.umu.se (J.C. Nieves), helena@cs.umu.se (H. Lindgren).

http://dx.doi.org/10.1016/j.ijar.2015.06.009
0888-613X/

In practice, knowledge bases are sometimes inconsistent, due to the presence of rules having exceptions, or because the available knowledge comes from several sources that do not necessarily agree [18], i.e., in sensor-based systems. *Extended logic programs* (ELP) [19] use both strong negation ¬ and negation-as-failure *not*, representing common-sense knowledge through logic programs. ELP deals directly with incomplete information instead of predicate calculus and circumscription, as its (axiom system) syntax and semantics are more complete and computationally superior [20]. ELP is not the only approach capturing incomplete information and exceptions; other non-monotonic logic approaches have also been introduced in the literature [21–23]. Two major semantics for ELP have been defined: (1) answer set semantics [19], an extension of *Stable model semantics*, and (2) a version of the Well-Founded Semantics (WFS) [24]. Answer set programming (ASP) is a form of declarative programming oriented towards difficult, primarily NP-hard, search problems [25]. ASP is based on the Stable model (answer sets) semantics to the analysis of negation as failure, but since WFS can be viewed as an efficient and skeptical approximation of Stable [26–28], it is treated as an approximation to answer set semantics.

In argumentation literature, there is an imbalance between theoretical research and pragmatic and experimental developments. Indeed, there is a limited availability of ABS formalisms with their implementations, i.e., showing a practical use in scenarios with incomplete and inconsistent information. Moreover, some of the ABS mentioned above fail to satisfy logical closure and consistency, which are important properties in any logical reasoning system [29,30]. Against this background, we introduce an *argumentation approach* for building arguments based on the answer set programming paradigm. Our approach takes a knowledge base captured by an extended logic program as input, obtaining as a result a set of *consistent* arguments, where an *argument* entails a relationship support-conclusion. In addition, we introduce a set of rationality postulates for ABS under extended logic programs, serving as a quality recommendation for logic-based argument systems, specifically in terms of logic programming approaches.

Essentially, our argumentation approach differs from previous approaches in: (1) our language for representing knowledge is based on extended logic programs [19], allowing us to deal naturally and conveniently with incomplete information; (2) we take advantage of the remarkable properties of answer set semantics, particularly a skeptical non-monotonic reasoning semantics: the WFS [24]; (3) the proposed rationality postulates are based on the well-known Gelfond–Lifschitz reduction [31], which is the core for defining the Stable model semantics; (4) Unlike other argumentation approaches to building arguments, which are mainly syntactic-based or proof-oriented, our approach is based on logic programming semantics inferences, i.e., Stable and WFS semantics.

WFS and Stable semantics capture the common-sense notion of *negation as failure*, leading to a natural treatment of exceptions. The *relevance* property, which is satisfied by WFS, allows us to identify only those clauses relevant to building a stratified support, inferring a conclusion and avoiding problems associated with the so-called *conflict propagation* [30] or *contamination* [32,33]. Furthermore, because WFS is polynomial time computable, we have developed an implementation, which, to our knowledge, is the first argumentation engine using this approach.

In summary, the following technical contributions are made:

- A notion of argument under the Well-Founded semantics and Stable semantics inference is proposed with a stratified program as support, using any extended logic program (stratified or not) as input.
- A set of rationality postulates for argument-based systems based on extended logic programs.
- A prototype tool which may be downloaded and tested.

Regarding to the first contribution, we show also that our characterization of an argument under WFS and Stable semantics coincides, by considering a stratified program as argument support. Moreover, since different well-acceptable semantics such as Completion semantics [34,35], Perfect model semantics [36,37] and PStable semantics [38] among others, also coincide with the stable semantics in the class of stratified programs, we can use our argument building approach with different logic programming semantics and obtain the same set of arguments.

The rest of the paper is structured as follows. In Section 2, we recall the theoretical background used throughout the paper. We introduce the argument and attack relationship concepts in Section 3 and we study some of the properties of these concepts. Rationality postulates for argumentation-based systems under ELP are explored in Section 4. Our argumentation tool and its architecture are described in Section 5. We present a discussion about the related work in the state-of-the-art in Section 6. In Section 7 a summary with our contributions and future work is presented. In Appendix A, the proofs of our formal result are presented.

2. Background

In this section some background about logic programs and argumentation semantics is introduced. We assume that the reader is familiar with basic terms of Logic Programming with *negation as failure*. A basic introduction about this topic can be found in [27].

2.1. Logic programs: syntax

Let us introduce the language of a propositional logic, which is constituted by propositional symbols: $p_0, p_1, \ldots$; connectives: $\wedge, \leftarrow, \neg,$ *not*, $\top$; and auxiliary symbols: (,), in which $\wedge, \leftarrow$ are 2-place connectives, $\neg,$ *not* are 1-place connectives