International Journal of Approximate Reasoning 58 (2015) 87–111

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

On the equivalence between logic programming semantics and argumentation semantics

Martin Caminada [a,*], Samy Sá [b,*], João Alcântara [b], Wolfgang Dvořák [c]

[a] University of Aberdeen, United Kingdom
[b] Universidade Federal do Ceará, Brazil
[c] University of Vienna, Faculty of Computer Science, Austria

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form 22 December 2014
Accepted 22 December 2014
Available online 29 December 2014

Keywords:
Abstract argumentation semantics
Logic programming semantics

ABSTRACT

In the current paper, we re-examine the connection between formal argumentation and logic programming from the perspective of semantics. We observe that one particular translation from logic programs to instantiated argumentation (the one described by Wu, Caminada and Gabbay) is able to serve as a basis for describing various equivalences between logic programming semantics and argumentation semantics. In particular, we are able to show equivalence between regular semantics for logic programming and preferred semantics for formal argumentation. We also show that there exist logic programming semantics (L-stable semantics) that cannot be captured by any abstract argumentation semantics.

1. Introduction

The connection between logic programming and formal argumentation goes back to logic programming inspired formalisms like the work of Prakken and Sartor [1] or the work of Simari and Loui [2], as well as to the seminal work of Dung [3] in which various connections were pointed out. To some extent, the work of Dung [3] can be seen as an attempt to provide an abstraction of certain aspects of logic programming. The connection between logic programming and argumentation is especially clear when it comes to comparing the different semantics that have been defined for logic programming with the different semantics that have been defined for formal argumentation. In the current paper, we continue this line of research. We do this by pointing out that one particular translation from logic programming to formal argumentation (the one of Wu et al. [4]) is able to account for a whole range of equivalences between logic programming semantics and formal argumentation semantics. This includes both existing results like the equivalence between stable model semantics (LP) and stable semantics (argumentation) [3], between well-founded semantics (LP) and grounded semantics (argumentation) [3], and between partial stable model semantics (LP) and complete semantics [4], as well as a newly proved equivalence between regular model semantics (LP) and preferred semantics (argumentation).

Our work is based on the fact that argumentation semantics are defined on the argument level, whereas logic programming semantics are defined on the conclusion level (with an argument being a defeasible derivation for a particular conclusion). Moreover, it holds that some of the most common argumentation semantics (grounded, preferred, semi-stable

* Corresponding authors.
E-mail addresses: martin.caminada@abdn.ac.uk (M. Caminada), samy@ufc.br (S. Sá), jnando@lia.ufc.br (J. Alcântara), wolfgang.dvorak@univie.ac.at (W. Dvořák).

http://dx.doi.org/10.1016/j.ijar.2014.12.004
0888-613X/

and stable) are based on selecting the complete labellings [5,6] where a particular label is maximal or minimal whereas some of the most common logic programming semantics (well-founded, regular, L-stable and stable) are based on selecting the P-stable models where a particular truth value is maximal or minimal. This, together with the previously observed equivalence between complete labellings and P-stable models [4] allows us to examine any additional equivalences in a systematic way: does maximizing or minimizing a particular label on the argument level coincide with maximizing or minimizing a particular truth value on the conclusion level?

The results of the current paper, however, are relevant for more than just the connection between logic programming and formal argumentation. They also shed light on specific aspects of instantiated argumentation theory in general (e.g. [7–9]). In particular, we show the connection between argument-labellings at the abstract level and conclusion-labellings at the instantiated level. With one notable exception, we are able to show that maximizing (or minimizing) a particular label (`in`, `out` or `undec`) at the argument level coincides with maximizing (or minimizing) the same label at the conclusion level. These results are relevant as they indicate the possibilities (and limitations) of applying argument-based abstractions to formalisms for nonmonotonic reasoning.

This paper is structured as follows. First, in Section 2, we introduce the main concepts to be applied in the current paper, such as the various semantics of abstract argumentation and logic programming to be examined. Then, in Section 3 we provide an overview of the three step process of instantiated argumentation and how it is applied in the particular context of logic programming based argumentation. In Section 4 we examine some existing work on the minimization and maximization of argument labellings. Similarly, in Section 5 we examine the issue of minimization and maximization of conclusion labellings. The connection between argument labellings and conclusion labellings is then studied in Section 6. We use this connection to study the equivalence between argumentation semantics and logic programming semantics in Section 7. For this, we point out that argument labellings coincide with argument extensions and conclusion labellings coincide with logic programming models. One notable exception on the equivalence between argumentation semantics and logic programming semantics is studied in Section 8, where we examine possible ways in which this equivalence can be restored. A reverse translation from argumentation frameworks to logic programs is then specified in Section 9, and it is observed that for this translation the equivalence of argumentation semantics and logic programming semantics is even stronger than for the translation of (unrestricted) logic programs to argumentation frameworks. Finally, we round off with a discussion of the obtained results in Section 10.

2. Preliminaries

In this section, we introduce the main definitions used throughout the paper as well as the first connections between formal argumentation and logic programming. We start with the definitions of abstract argumentation frameworks and their various semantics and then move on to logic programs and their various semantics. In order to highlight similarities between these concepts, we provide definitions of each formalism in a similar fashion.

2.1. Abstract argumentation frameworks and semantics

In the current paper, we follow the approach of Dung [3]. To simplify things, we restrict ourselves to finite argumentation frameworks.

Definition 1. (See [3].) An *argumentation framework* is a pair (Ar, att) where Ar is a finite set of arguments and $att \subseteq Ar \times Ar$.

Arguments are related to others by the attack relation att, in the sense that an argument A attacks the argument B iff $(A, B) \in att$. An argumentation framework can be depicted as a directed graph where the arguments are nodes and each attack is an arrow.

Definition 2 *(Defense/conflict-free).* (See [3].) Let (Ar, att) be an argumentation framework, $A \in Ar$ and $\mathcal{A}rgs \subseteq Ar$.

- $\mathcal{A}rgs$ is said to be *conflict-free* iff there exists no arguments $A, B \in \mathcal{A}rgs$ such that $(A, B) \in att$.
- $\mathcal{A}rgs$ is said to *defend* an argument A iff every argument that attacks A is attacked by some argument in $\mathcal{A}rgs$.
- The *characteristic function* $F : 2^{Ar} \to 2^{Ar}$ is defined as $F(\mathcal{A}rgs) = \{A | A$ is defended by $\mathcal{A}rgs\}$.
- A conflict-free set $\mathcal{A}rgs$ is said to be *admissible* iff $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$, that is the arguments in the set can defend themselves against any attackers in the framework.
- We write $\mathcal{A}rgs^+ = \{A | A$ is attacked by an argument in $\mathcal{A}rgs\}$ to refer to the set of arguments attacked by $\mathcal{A}rgs$.

The traditional approaches to argumentation semantics are based on sets (commonly referred to as “extensions”) of arguments. Some of the mainstream approaches are summarized in the following definition.[1]

[1] The characterization of the extension-based semantics in Definition 3 is slightly different than the way these were originally defined, by Dung [3], but equivalence is proved by Caminada and Gabbay [6].