Argumentation Inference as Logic Programming Inference: An Extended Abstract

Juan Carlos NIEVES
Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden
jcnieves@cs.umu.se

Abstract

In this extended abstract, we present a small survey of recent results in the study of argumentation theory as logic programming. In particular, we survey recent characterizations of eight argumentation semantics, *i.e.*, *Ground, Stable, Preferred, Complete, Semi-stable, Ideal, CF2, Stage*, as logic programming semantics with negation as failure.

These results consolidate the strong relationship between argumentation semantics and logic programming semantics with negation as failure.

1 Introduction

During the last decade research in argumentation has been rapidly increased from both theoretical and practical points of view. These research activities have been motivated by the increased interest of building intelligent systems able to interact with other agents (software or humans) in an autonomous way. For instance, the theoretical results from argumentation have coined the so called *agreement technologies* (ATs)[1]. ATs refer to computer systems in which autonomous software agents negotiate with one another, typically on behalf of humans, in order to come to mutually acceptable agreements.

Theoretical argumentation research has been strongly influenced by the abstract argumentation theory of Dung [8]. This approach is mainly oriented towards managing the interaction between arguments. Argumentation has been regarded as a non-monotonic reasoning approach since it was suggested as an inference reasoning approach. Dung also showed that argumentation inference can be regarded as a logic programming inference with *negation as failure* [8]. In his seminar paper [8], Dung introduced four argumentation semantics: *grounded, stable, preferred* and *complete* semantics; moreover, he showed that both the grounded and stable can be regarded as logic programming inference by considering the well-founded [12] and stable model [13] semantics, respectively.

Even though, currently, Dung's argumentation semantics are well accepted, they can exhibit a variety of problematic behaviors: *Emptiness*, *Non-existence* and *Multiplicity* [3]. Therefore, following Dung's argumentation style, several new argumentation semantics have been proposed. Among them, *Semi-stable, Ideal, CF2 and Stage* semantics have been explored from different points of view [1]. Indeed, these four argumentation semantics have been also characterized by different logic programming semantics (see Section 4).

The aim of this small survey is to resume recent results in the characterization of argumentation semantics in terms of logic programming semantics. We argue that these characterizations give place to *new research threads* which must be explored in order to pursue a better understanding of the argumentation inference. This understanding can impact in, at least: 1.- the definition of new algorithms for implementing argumentation systems, 2.- the definition of new criteria for comparing argumentation semantics from the non-monotonic reasoning point of view. It is worth mentioning that the computational complexity of the decision prob-

[1] http://www.agreement-technologies.eu/

lems of argumentation semantic has been shown to range from NP-complete to Π_2^p [10]. Therefore, to identify efficient algorithms for implementing argumentation systems is high valuable. Moreover, to identify different interpretations of the argumentation semantics can give place to new criteria for comparing argumentation semantics. To identify criteria for comparing argumentation semantics is quite relevant nowadays; given that, new argumentation semantics are still appearing.

The rest of the paper is split as follows: In Section 2, we introduce the definition of some argumentation semantics. In Section 3, we resume the main mappings which have been used for mapping argumentation frameworks into logic programs. In Section 4, we resume the characterizations of argumentation semantics in terms of logic programming semantics which have been done. In the last section, we present a small outline of conclusions.

2 Argumentation Semantics

In this section, we introduce the definition of some argumentation semantics mainly *stable, grounded, preferred, complete, ideal, semi-stable and stage semantics*. To this end, we start defining the basic structure of an argumentation framework.

Definition 1 *[8] An argumentation framework is a pair $AF := \langle AR, attacks \rangle$, where* AR *is a finite set of arguments, and* attacks *is a binary relation on* AR, *i.e.,* attacks $\subseteq AR \times AR$.

We say that *a attacks b* (or b is attacked by a) if $attacks(a, b)$ holds. Similarly, we say that a set S of arguments attacks b (or b is attacked by S) if b is attacked by an argument in S.

Let us observe that an argumentation framework is a simple structure which captures the conflicts of a given set of arguments. In order to select *coherent points of view* from a set of conflicts between arguments, Dung introduced a set of *patterns of selection of arguments*. These patterns of selection of arguments were called *argumentation semantics*. Dung defined his argumentation semantics based on the basic concept of *admissible set*:

Definition 2 *[8]*

- *A set S of arguments is said to be conflict-free if there are no arguments a, b in S such that a attacks b.*
- *An argument $a \in AR$ is acceptable with respect to a set S of arguments if and only if for each argument $b \in AR$: If b attacks a then b is attacked by S.*
- *A conflict-free set of arguments S is admissible if and only if each argument in S is acceptable w.r.t. S.*

Let us introduce some notation. Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. $S^+ = \{b | a \in S$ and $(a, b) \in attacks\}$.

Definition 3 *[1, 8, 9] Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. An admissible set of argument $S \subseteq AR$ is:*

- stable *if and only if S attacks each argument which does not belong to S.*
- preferred *if and only if S is a maximal (w.r.t. inclusion) admissible set of AF .*
- ideal *if and only if it is contained in every preferred extension of AF.*
- complete *if and only if each argument, which is acceptable with respect to S, belongs to S.*
- *the* grounded *extension of AF if and only if S is the minimal (w.r.t. inclusion) complete extension of AF*[2].
- semi-stable *if and only if S is a complete extension such that $S \cup S^+$ is maximal w.r.t. set inclusion.*

In addition to the argumentation semantics based on admissible sets, in the state of art, there are other approaches for defining argumentation semantics, *e.g.*, [2, 21]. One of these approaches is the approach based on *conflict-free sets*. In this setting, we can point out: *Stage Semantics* [21] and *CF2* [2]. Stage semantics is defined as follows:

[2] This is not the original definition of the grounded semantics introduced in [8]; however, it was show that the grounded extension can be defined in terms of complete extensions [1].

Definition 4 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. E is a stage extension iff E is a conflict free set and $E \cup E^+$ is maximal w.r.t. set inclusion.*

As we can observe the definition of stage semantics is similar to the definition of semi-stable semantics. The only difference, between them, is that one considers complete extensions and the other one considers conflict free sets. Nevertheless, stage semantics and semi-stable represent different patterns of selection of arguments.

Due to lack of space, we skip to present the definition of CF2; however, let us observe that CF2 is based on the graph theoretical notion of *strongly connected components* (SCC) [2]. CF2 was suggested as one of the possible approaches for dealing with some problems of Dung's semantics, *e.g.*, emptiness. We refer to the reader to [2] for a detailed presentation of CF2.

3 Mappings from argumentation frameworks to normal programs

The usual way for studying argumentation as logic programming is to map argumentation frameworks into logic programs (also called logical theories). Currently, we can find different mappings of argumentation frameworks into logical theories. Dung introduces the following basic meta-interpreter (or mapping) in terms of logic programs with negation as failure [8]:

$$P_{AF}^{Dung}: \quad \begin{array}{l} acc(X) \leftarrow not\ def(X). \\ def(X) \leftarrow attack(Y,X), acc(Y). \end{array}$$

where $acc(X)$ stands for argument X is *acceptable* and $def(X)$ stands for argument X is *defeated*.

By using P_{AF}^{Dung}, Dung characterized the grounded and the stable semantics in terms of the well-founded and stable model semantics respectively. This mapping basically is the first mapping which was suggested for regarding argumentation frameworks as logic programs.

There is a second mapping which has been explored in order to map an argumentation framework into a normal logic program: Given an argumentation framework $AF := \langle AR, attacks\rangle$:

$$P_{AF} = \bigcup_{x \in AR} \{x \leftarrow \bigwedge_{(y,x) \in attacks} not\ y\}$$

This mapping was first introduced in [17] in order to show that the answer sets of P_{AF} corresponds to the stable extensions of AF (see Theorem 1 of [17]). In [22], the authors showed that the complete semantics can be characterized in terms of the 3-valued stable semantics and P_{AF}. This mapping was also explored by Gabbay in order to map argumentation frameworks into logic programs [11].

From the declarative point of view, the mapping P_{AF} only specifies a basic specification of *why an argument can belong to an extension of an argumentation semantics*. Indeed, like Dung's mapping, P_{AF} only is capturing the idea of *conflict-freeness.*

In [15], a pair of mappings were introduced. These mappings are defined as follows:

Definition 5 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework and $a \in AR$. We define a pair of transformation functions:*

$$\Pi^-(a) = \bigcup_{b:(b,a) \in attacks} \{def(a) \leftarrow \ not\ def(b)\}$$

$$\Pi^+(a) = \bigcup_{b:(b,a) \in attacks} \{def(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} def(c)\}$$

Let us see that if a given argument a has no attacks, $\Pi^-(a) = \{\}$ and $\Pi^+(a) = \{\}$. This situation happens because any argument that has no attacks is an acceptable argument which means that it belongs to all admissible sets of AF.

By considering $\Pi^-(a)$ and $\Pi^+(a)$, a couple of mappings from argumentation frameworks into logic programs can be defined.

Definition 6 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. We define their associated normal programs as follows:*

$$\Pi_{AF}^- := \bigcup_{a \in AR} \{\Pi^-(a)\}$$

$$\Pi_{AF} := \Pi_{AF}^- \cup \bigcup_{a \in AR} \{\Pi(a)^+\}$$

It is obvious that Π_{AF}^- is a subset of Π_{AF}. However, each mapping is capturing different concepts: Π_{AF}^- is basically a declarative specification of the idea of conflictfreeness and Π_{AF} is basically a declarative specification of the ideas of conflict-freeness and reinstatement. Indeed, one can see that the 2-valued logical models of Π^-_{AF} characterize the conflict free sets of AF and the 2-valued logical models of Π_{AF} characterize the admissible sets of AF.

Even though, Π^-_{AF} syntactically looks different from Dung's mapping (P^{Dung}_{AF}); basically, Π^-_{AF} is the grounded program of P^{Dung}_{AF}[3]. In this setting, we can say that Π_{AF} basically is adding new constraints to Dung's mapping. Hence, Π_{AF} is a conservative extension of Dung's mapping.

4 Characterization of argumentation semantics as logic programming inferences

In the literature, there are different characterizations of argumentation semantics as logic programming semantics. A summary of these characterizations is presented in Table 1

The first observation from Table 1 is that the five argumentation semantics suggested by Dung, *et al.*, grounded, stable, preferred, complete and ideal semantics, have been characterized by different logic programming semantics. Moreover, other argumentation semantics which follow Dung's argumentation semantics style, such as semi-stable, CF2 and Stage Semantics, have been also characterized by different logic programming semantics. In this setting, we can argue that any argumentation semantics must be characterized by a logic programming semantics. To find a logic programming semantics which could characterize a given argumentation semantics cannot be a big deal; however, an interesting question of a given characterization of an argumentation semantics in terms of a logic programming semantics is: *is such logic programming semantics interesting or well known?* By *interesting semantics*, we mean such semantics satisfies well-expected properties of non-monotonic reasoning as the properties suggested by Dix [6, 7]. For instance, Dix showed that WFS and WFS$^+$ are two well-behaved semantics[4]. Therefore, WFS and WFS$^+$ argue for the well behavior, as *non-monotonic reasoning inferences*, of the grounded and ideal semantics, respectively. By *well known semantics*, we mean a logic programming semantics which have been studied by different authors. For instance the Clark's completion semantics is an old logic programming semantics which has been influencing modern logic programming semantics such as the stable model semantics. Hence, to find logic programming semantics, which have been studied by different authors, can suggest new interpretations of a given argumentation semantics.

Another important observation from Table 1 is that there is not a unique mapping from argumentation frameworks into logic programs in order to regard argumentation as logic programming. We cannot argue that one mapping is better than another; however, we can observe that except by the characterizations of the grounded and the stable semantics, which have been characterized by stable models and Well-Founded semantics by using different mappings, each mappings, of Table 1, is using different logic programming semantics for characterizing a particular argumentation semantics. For instance, the mapping P_{AF} characterizes the complete semantics by using the 3-valued stable model semantics; on the other hand, the mapping Π_{AF} is using the Clark's completion semantics for characterizing the complete semantics. In this setting, we can argue that we need some criteria for deciding when a mapping from an argumentation framework into a logic programs is a *suitable codification* for studying argumentation as logic programming. It will not be strange that new mappings could appear; however, to find consistent mappings from an argumentation framework into a logic program can standardize the interpretation of an argumentation framework as a logic program. As we observed in Section 3, each of the mappings, explored until now, can be regarded as declarative specifications of either conflict-free sets or admissible sets. Hence the scope of each mapping is strongly related to the concept which is capturing. For instance, Π_{AF} is a declarative specification of admissible sets; moreover, observing Table 1, we can see that Π_{AF} has been a useful

[3]For obtaining Π^-_{AF} from P^{Dung}_{AF}, we need to turn grounded P^{Dung}_{AF} and apply partial evaluation.

[4]The interesting reader can find in [7] the formal definition of a well-behaved logic programming semantics.

Table 1: Characterization of argumentation semantics as logic programming inferences.

Argumentation semantics	**Logic programming semantics using P_{AF}**	**Logic programming semantics using Π_{AF}**	**Logic programming semantics using Π^{-}_{AF}**
Grounded Semantics	Well-founded semantics [4], the Kripke-Kleene model [20]	Well-founded semantics [5]	Well-founded semantics [8], the Kripke-Kleene model [20]
Stable Semantics	Stable model semantics [4, 17], Supported models [20]	Stable model semantics [5]	Stable models semantics [8], Supported models [20]
Preferred Semantics	Regular semantics [4], M-supported models, M-stable models [20]	P-stable Semantics [5]	M-supported models, M-stable models [20]
Complete Semantics	3-valued stable semantics [22, 20], 3-valued supported models [20]	Supported Models [19]	3-valued stable semantics, 3-valued supported models [20]
Semi-stable Semantics	L-Stable [4, 20], L-Supported models [20]	GL-supported models [18]	L-supported models, L-stable models [20]
Ideal Semantics		WFS^{+} [14]	
CF2 Semantics			MM^{*} [16]
Stage Semantics			GL-stage models [18]

mapping for characterizing the six well-acceptable admissibility-based argumentation semantics. On the other hand, P_{AF} is a declarative specification of conflict-free sets and, observing Table 1, P_{AF} has been a useful mapping for characterizing five admissibility-based argumentation semantics. Π^{-}_{AF} is a really interesting mapping since it has been able to characterize both admissibility-based argumentation semantics and conflict-free-based argumentation semantics. It could be interesting to see if there exists a logic programming semantics which could characterize ideal semantics by using either P_{AF} or Π^{-}_{AF}. Moreover, to see if the given logic programming semantics could suggest a different interpretation, *w.r.t.* WFS^{+}, of the ideal semantics as a non-monotonic reasoning inference.

So far, we can observe that understanding argumentation semantics from a logic programming point of view depends mainly on two variables: 1.- The logic programming semantics which infers the given argumentation semantics and 2.- The declarative specification of an argumentation framework in terms of logic programs. These two variables can suggest different interpretations of a given argumentation semantics; even more, to suggest an approach for exploring new argumentation semantics as the approach explored in [16].

5 Conclusions

Argumentation inference is strongly influenced by Dung's argumentation style. Since Dung's approach was introduced, it has been showed that this approach can be regarded as logic programming inference. Currently, most of the well acceptable argumentation semantics have been characterized as logic programming inference. This evidence argues that whenever a new argumentation semantics appears, it is totally reasonable to ask to be characterized as logic programming inference. However, to use arbitrary logic programming semantics for characterizing a given argumentation semantics cannot be valuable if the given logic programming semantics is not interesting or well known. Moreover, the introduction of new mappings of argumentation frameworks into logic programs must be motivated by properties such as the number of argumentation semantics able to characterize.

References

[1] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[2] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168:162–210, October 2005.

[3] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[4] Martin Caminada, Samy Sá, and João Alcântara. On the equivalence between logic programming semantics and argumentation semantics. Rechnical Report ABDN-CS-13-01, University of Aberdeen, 2013.

[5] José Luis Carballido, Juan Carlos Nieves, and Mauricio Osorio. Inferring Preferred Extensions by Pstable Semantics. *Iberoamerican Journal of Artificial Intelligence (Inteligencia Artificial) ISSN: 1137-3601, (doi: 10.4114/ia.v13i41.1029)*, 13(41):38–53, 2009.

[6] Jürgen Dix. A classification theory of semantics of normal logic programs: I. strong properties. *Fundam. Inform.*, 22(3):227–255, 1995.

[7] Jürgen Dix. A classification theory of semantics of normal logic programs: II. weak properties. *Fundam. Inform.*, 22(3):257–288, 1995.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[9] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[10] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

[11] Dov M. Gabbay and Artur S. d'Avila Garcez. Logical modes of attack in argumentation networks. *Studia Logica*, 93(2-3):199–230, 2009.

[12] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.

[13] Michael Gelfond and Vladimir Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[14] Juan Carlos Nieves and Mauricio Osorio. Ideal extensions as logical programming models. *Journal of Logic and Computation*, DOI:10.1093/logcom/exu014, 2014.

[15] Juan Carlos Nieves, Mauricio Osorio, and Ulises Cortés. Preferred Extensions as Stable Models. *Theory and Practice of Logic Programming*, 8(4):527–543, July 2008.

[16] Juan Carlos Nieves, Mauricio Osorio, and Claudia Zepeda. A Schema for Generating Relevant Logic Programming Semantics and its Applications in Argumentation Theory. *Fundamenta Informaticae*, 106(2-4):295–319, 2011.

[17] Juan Carlos Nieves, Mauricio Osorio, Claudia Zepeda, and Ulises Cortés. Inferring acceptable arguments with answer set programming. In *Sixth Mexican International Conference on Computer Science (ENC 2005)*, pages 198–205. IEEE Computer Science Press, September 2005.

[18] Mauricio Osorio and Juan Carlos Nieves. Semi-stable and stage extensions as 2-valued models. *Paper under evaluation.*

[19] Mauricio Osorio, Juan Carlos Nieves, and Alejandro Santoyo. Complete Extensions as Clark's Completion Semantics. In *Mexican International Conference on Computer Science*, pages 81–88. IEEE Computer Science Press, 2013.

[20] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.

[21] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC 1996)*, 1996.

[22] Yining Wu, Martin Caminada, and Dov M. Gabbay. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica*, 93(2-3):383–403, 2009.