doi:10.1017/S1471068419000139

Paracoherent Answer Set Semantics meets Argumentation Frameworks

GIOVANNI AMENDOLA and FRANCESCO RICCA
University of Calabria, Rende, Italy
(*e-mail:* {amendola,ricca}@mat.unical.it)

submitted 23 July 2019; accepted 31 July 2019

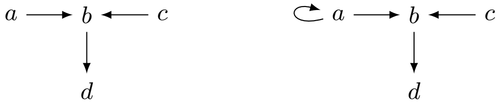
Abstract

In the last years, abstract argumentation has met with great success in AI, since it has served to capture several non-monotonic logics for AI. Relations between argumentation framework (AF) semantics and logic programming ones are investigating more and more. In particular, great attention has been given to the well-known stable extensions of an AF, that are closely related to the answer sets of a logic program. However, if a framework admits a small incoherent part, no stable extension can be provided. To overcome this shortcoming, two semantics generalizing stable extensions have been studied, namely semi-stable and stage. In this paper, we show that another perspective is possible on incoherent AFs, called paracoherent extensions, as they have a counterpart in paracoherent answer set semantics. We compare this perspective with semi-stable and stage semantics, by showing that computational costs remain unchanged, and moreover an interesting symmetric behaviour is maintained.

KEYWORDS: Answer Set Programming, Abstract Argumentation, Paracoherent reasoning

1 Introduction

In the last years, abstract argumentation theory (Dung 1995) has met with great success in AI (Bench-Capon and Dunne 2007), since it has served to capture several non-monotonic logics for AI. Recently, relations between abstract argumentation semantics and logic programming semantics has been studied systematically in (Strass 2013; Caminada et al. 2015). These can be highlighted by using a well-known tool for translating Argumentation Frameworks (AFs) to logic programs (Wu et al. 2009). In particular, given an AF F one can build the corresponding logic program, P_F as follows: For each argument a in F, if b_1, b_2, ..., b_m is the set of its defeaters, construct the rule $a \leftarrow not\ b_1,\ not\ b_2,\ \ldots,\ not\ b_m$. Intuitively, each of these rules means that an argument is accepted (inferred as true) if, and only if, all of its defeaters are rejected (false) for some reason. It is known that the well-founded model of P_F (Gelder et al. 1991) corresponds to the grounded extension of F (Dung 1995); stable models (or answer sets) of P_F (Gelfond and Lifschitz 1991) correspond to stable extensions of F (Dung 1995); regular models (You and Yuan 1994) correspond to preferred extensions; P-stable models (Przymusinski 1991b) correspond to complete extensions (Wu et al. 2009); and L-stable models (Eiter et al. 1997) correspond to semi-stable extensions (Caminada et al. 2015). Focusing on the stable semantics, we recall that a stable extension is defined as a conflict-free set (i.e., no argument in the set attacks another one in the set) that



(a) A stable extension. (b) No stable extension.

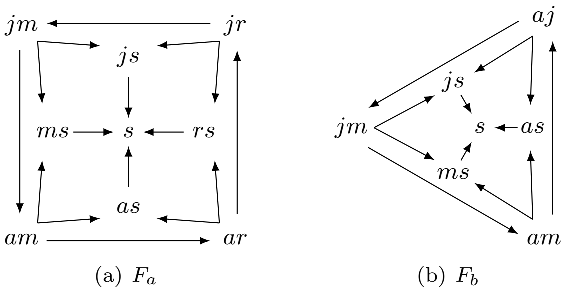
Fig. 1. Examples of argumentation frameworks.

attacks every other argument outside of it (Dung 1995). In particular, the labelling-approach (Jakobovits and Vermeir 1999) implies that no argument is labelled as undecidable (i.e., an argument will be either true or false). This condition yields very solid solutions. For instance, consider the AF reported in Figure 1(a). As both arguments a and c are attacked by no other argument, they will be true; so that, b will be false (it is attacked by a and c); and, finally, d will be true, as it is attacked by b only. Hence, $\{a, c, d\}$ is a stable extension. However, this solidity is given at the price of not offering any solution in many situations. In particular, if the AF includes even a small part admitting no stable extension, no stable extension can be provided for the entire AF. This is explained by saying that this semantics is not "crash resistant" (Caminada et al. 2012). For instance, if we consider the AF reported in Figure 1(b), that differs from the previous one only for an attack from a to itself, then no stable extension exists.

To overcome this shortcoming, argumentation semantics that extend the stable one have been proposed. More specifically, they coincide with the stable semantics, whenever a stable extension exists; and provide others solutions, whenever no stable extension exists. Currently, there are two semantics with these properties: the Stage semantics (Verheij 1996) and the Semi-stable semantics (Caminada 2006). A stage extension is a conflict-free set of arguments A, such that $A \cup A^+$ is (subset) maximal with respect to conflict-free sets, where A^+ denotes the set of all arguments attacked by an argument of A. A semi-stable extension is a conflict-free set A, where each argument attacking an argument of A, is in turn attacked by an argument of A, such that $A \cup A^+$ is maximal. By considering the AF reported in Figure 1(b), the set $\{c, d\}$ is both a stage extension and a semi-stable extension. However, in general, stage semantics and semi-stable semantics are very different (Caminada et al. 2012).

In this paper, we provide an alternative view on how to generalize the stable semantics, based on paracoherent semantics introduced in Answer Set Programming (ASP) (Brewka et al. 2011). Actually, there are two main paracoherent semantics for logic programs. The first was introduced by (Sakama and Inoue 1995), and it is known as *semi-stable model semantics* (not to be confused with semi-stable semantics in argumentation); while the most recent was introduced by (Eiter et al. 2010) to avoid some anomalies concerning modal logic properties, and it is known as *semi-equilibrium model semantics*. As we will see, these two semantics coincide in our settings, and we refer to them as *paracoherent answer sets*.

A distinctive property of the new paracoherent semantics for AF is a symmetric behaviour, that is not present in stage and semi stable semantics. To exemplify it, we use a concrete classic application of AF considered orginally by (Dung 1995), namely the *Stable Roommates Problem* (SRP) (Amendola 2018). SRP is a matching problem with several variants and real-word applications, where persons have to be matched while respecting their preferences (Gale and Shapley 1962). Suppose that Mark (m) prefers John (j) to



(a) F_a (b) F_b

Fig. 2. Argumentation frameworks for SRP.

Annie (a), and Annie to Shrek (s); John prefers Robert (r) to Mark, and Mark to Shrek; Robert prefers Annie to John, and John to Shrek; Annie prefers Marc to Robert, and Robert to Shrek; while Shrek wants to stay alone. The AF F_a in Figure 2(a) models the problem using arguments for possible matchings that attack each other according to preferences. Note that F_a admits two stable extensions (i.e., matchings). One solution pairs John and Mark (jm), Annie and Robert (ar), while Shrek stays alone; and the other solution pairs Annie and Mark (am), John and Robert (jr), again Shrek stays alone. Hence "Shrek is alone" is a skeptical argument, as expected. Consider now the AF F_b in Figure 2(b) with one less person and the following preferences: Mark prefers John to Annie, and Annie to Shrek; John prefers Annie to Mark, and Mark to Shrek; Annie prefers Mark to John, and John to Shrek; again Shrek wants to stay alone. Despite the AF F_b has the same symmetric structure of F_a, it has no stable extension (due to odd-length cycles). This means that the SRP problem has no solution, and, in practice, one has to accept a non comfortable solution (i.e., a non stable one). Pragmatically, one can minimizes the number of persons that have no roommate, and always keep Shrek (disliking all mates) alone, as in the stable case. Unluckily, the semi-stable semantics cannot suggest a practical solution to the problem (as the unique admissible set is the empty one); on the other hand, it can be verified that in stage extensions Shrek is always paired. Thus, both semantics extending the stable one do not behave as in F_a, where Shrek is always alone and persons that have no roommate are minimized, i.e., we would expect to pair Mark with John or John with Annie or Annie with Mark. This symmetric behavior is kept in the semantics introduced here. As argued in (Baroni et al. 2005) "it is counter-intuitive that different results in conceptually similar situations depend on the length of the cycle: symmetry reasons suggest that all cycles should be treated equally and should yield the same results". According to this observation, our semantics provides an approach that is more adherent to this ideal behavior than related proposals. Moreover, paracoherent extensions could represent plasible scenarios in several real world applications of AFs where odd cycles naturally appear, such as legal reasoning (Prakken and Sartor 1996; Verheij 2003; Gabbay 2016), dialog and negotiation (Amgoud et al. 2000), planning (Pollock 1998), and traffic networks (Gaggl et al. 2018).

In the paper, we define the *paracoherent extensions* for AFs that are based on the concept of *stabilizer*. Stabilizers capture what is missing to a set of arguments to become stable. Next, we show that paracoherent extensions can be considered as generalization of the stable extensions, just like the semi-stable and the stage ones. Then, by using the direct translation from AFs to logic programs, we show a correspondence between paracoherent extensions and paracoherent answer sets. Moreover, we prove that credulous

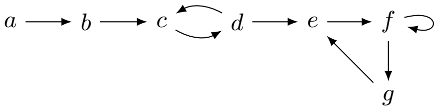


Fig. 3. Argumentation framework of Examples 2, 5, 6, 7.

and skeptical reasoning for paracoherent semantics are Σ_2^P-complete and Π_2^P-complete, respectively (as for semi-stable and stage (Dunne and Wooldridge 2009; Dunne and Caminada 2008; Dvorák and Woltran 2010)). Finally, we discuss and relate the paracoherent semantics with several existing ones, showing that paracoherent semantics has an interesting symmetric behaviour on graphs with odd-length cycles.

2 Preliminaries

We overview basic concepts in argumentation and paracoherent answer set semantics.

2.1 Abstract Argumentation Semantics

An AF F is a pair (Ar, att), where Ar is a finite set of *arguments*, and $att \subseteq Ar \times Ar$ is a set of *attacks*. Hence, an AF can be represented by a directed graph, where nodes are arguments, and edges are attacks. For instance, concerning the AF in Figure 1(a), we have $Ar = \{a, b, c, d\}$, and $att = \{(a, b), (c, b), (b, d)\}$. Let $A \subseteq Ar$ be a set of arguments. We denote by A^+ the set of all arguments in Ar attacked by an argument in A, i.e. $A^+ = \{b \in Ar \mid (a, b) \in att$, and $a \in A\}$. Then, A is *conflict-free* in F if, for each $a, b \in A$, $(a, b) \notin att$; A is *admissible* in F if A is conflict-free, and, given $a \in A$, for each $b \in Ar$ with $(b, a) \in att$, there is $c \in A$ such that $(c, b) \in att$ (say that a is *defended* by A in F); A is *complete* in F if A is admissible in F and each $a \in Ar$ defended by A in F is contained in A; A is a *stable extension* if $A^+ = Ar \setminus A$ (i.e., A is conflict-free, and for each $a \in Ar \setminus A$, there is $b \in A$, such that $(b, a) \in att$); A is a *semi-stable extension* if A is complete and $A \cup A^+$ is maximal (hereafter, we write just maximal for maximal w.r.t. subset inclusion); A is a *stage extension* if A is conflict-free and, for each B conflict-free, $A \cup A^+ \not\subset B \cup B^+$ (i.e., $A \cup A^+$ is maximal with respect to conflict-free sets). We denote by $cf(F)$, $adm(F)$, $comp(F)$, $stb(F)$, $sem(F)$, $stage(F)$, the sets of all conflict-free, admissible, complete, stable, semi-stable, and stage extensions, respectively.

Example 1
Let $F = (\{a, b, c, d\}, \{(a, b), (c, b), (b, d)\}$ be the AF reported in Figure 1(a). We have $cf(F) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{c, d\}, \{a, c, d\}\}$; $adm(F) = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{a, d\}, \{c, d\}, \{a, c, d\}\}$; $comp(F) = \{\{a, c, d\}\}$; $stb(F) = sem(F) = stage(F) = \{\{a, c, d\}\}$.

It is known that $cf(F) \supseteq adm(F) \supseteq comp(F) \supseteq sem(F) \supseteq stb(F)$; while $cf(F) \supseteq stage(F) \supseteq stb(F)$. Moreover, both semi-stable and stage semantics coincide with the stable one, whenever a stable extension exists; and provide others solutions, whenever no stable one exists (Caminada et al. 2012). The main difference between the two semantics is that, on the one hand, a stage extension is a maximal conflict-free set, while in general the semi-stable extension is not; on the other hand, a semi-stable extension is an admissible extension, while in general the stage extension is not.

Example 2
The AF reported in Figure 3 has no stable exstension, while it has a unique semi-stable extension, $\{a, d\}$; and 3 stage ones $\{a, c, e\}$, $\{a, c, g\}$, and $\{a, d, g\}$.

Moreover, we also recall the following decision problems corresponding to reasoning tasks on AFs. Given an AF F, a semantics σ, and an argument a, decide whether a: (1) is contained in some σ-extension of F (credulous reasoning); (2) is contained in all σ-extensions of F (skeptical reasoning). It is known that credulous reasoning is NP-complete for stable semantics, and Σ_2^P-complete for semi-stable and stage semantics; while skeptical reasoning is coNP-complete for stable semantics, and Π_2^P-complete for semi-stable and stage semantics (Dunne and Wooldridge 2009; Dunne and Caminada 2008; Dvorák and Woltran 2010).

2.2 Paracoherent Answer Set Semantics

We concentrate on logic programs over a propositional signature Σ. A *rule* r is of the form

$$a_1 \vee \ldots \vee a_l \leftarrow b_1, \ldots, b_m, not\ c_1, \ldots, not\ c_n \tag{1}$$

where all a_i, b_j and c_k are atoms (from Σ); $l, m, n \geq 0$, and $l + m + n > 0$; *not* represents *default negation*. The set $H(r) = \{a_1, ..., a_l\}$ is the *head* of r, while $B^+(r) = \{b_1, ..., b_m\}$ and $B^-(r) = \{c_1, \ldots, c_n\}$ are the *positive body* and the *negative body* of r, respectively; the *body* of r is $B(r) = B^+(r) \cup B^-(r)$. If $B(r) = \emptyset$, we then omit $\leftarrow$; and if $|H(r)| \leq 1$, then r is *normal*. A *program* P is a finite set of rules. P is called *normal* if each $r \in P$ is normal.

Any set $I \subseteq \Sigma$ is an *interpretation*. I is a *model* of a program P (denoted $I \models P$) iff for each rule $r \in P$, $I \cap H(r) \neq \emptyset$ whenever $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$ (denoted $I \models r$). A model M of P is *minimal* iff no model $M' \subset M$ of P exists. Given an interpretation I, we denote by P^I the well-known *Gelfond-Lifschitz reduct* (Gelfond and Lifschitz 1991) of P w.r.t. I, that is the set of rules $a_1 \vee ... \vee a_l \leftarrow b_1, ..., b_m$, obtained from rules $r \in P$ of form (1), such that $B^-(r) \cap I = \emptyset$. A model M of P is called *answer set* (or *stable model*) of P, if M is a minimal model of P^M. We denote by $AS(P)$ the set of all answer sets of P. Next, we introduce two paracoherent semantics. The first one is known as *semi-stable model semantics* (Sakama and Inoue 1995). We consider an extended signature $\Sigma^\kappa = \Sigma \cup \{Ka \mid a \in \Sigma\}$. Intuitively, Ka can be read as a is believed to hold. The semi-stable models of a program P are obtained from its *epistemic κ-transformation.*

Definition 1 (*Epistemic κ-transformation P^κ*)
Let P be a program. Then its epistemic κ-transformation is defined as the program P^κ obtained from P by replacing each rule r of the form (1) in P, such that $B^-(r) \neq \emptyset$, with:

$$\lambda_{r,1} \vee \ldots \vee \lambda_{r,l} \vee Kc_1 \vee \ldots \vee Kc_n \leftarrow b_1, \ldots, b_m; \tag{2}$$

$$a_i \leftarrow \lambda_{r,i}; \tag{3}$$

$$\leftarrow \lambda_{r,i}, c_j; \tag{4}$$

$$\lambda_{r,i} \leftarrow a_i, \lambda_{r,k}; \tag{5}$$

for $1 \leq i, k \leq l$ and $1 \leq j \leq n$, where $\lambda_{r,i}$ are fresh atoms.

Given an interpretation I^κ over $\Sigma' \supseteq \Sigma^\kappa$, let $\mathscr{G}(I^\kappa) = \{Ka \in I^\kappa \mid a \notin I^\kappa\}$ denote the atoms believed true but not assigned true, also referred to as the gap of I^κ. Given a set $\mathscr{F}$ of interpretations over Σ', an interpretation $I^\kappa \in \mathscr{F}$ is *maximal canonical in* $\mathscr{F}$, if no $J^\kappa \in \mathscr{F}$ exists such that $\mathscr{G}(I^\kappa) \supset \mathscr{G}(J^\kappa)$. By $mc(\mathscr{F})$ we denote the set of maximal canonical interpretations in $\mathscr{F}$. Semi-stable models are defined as maximal canonical interpretations among the answer sets of P^κ, and the set of all semi-stable models of P is denoted by $SST(P)$, i.e., $SST(P) = \{S \cap \Sigma^\kappa \mid S \in mc(AS(P^\kappa))\}$.

The second one is called *semi-equilibrium model semantics* and was introduced by (Eiter et al. 2010) to amend anomalies in semi-stable model semantics. Semi-equilibrium models may be computed as maximal canonical answer sets of an extension of the epistemic κ-transformation.

Definition 2 (*Epistemic HT-transformation* P^{HT})
Let P be a program over Σ. Then its epistemic HT-transformation P^{HT} is defined as the union of P^κ with the set of rules:

$$Ka \leftarrow a, \tag{6}$$

$$Ka_1 \vee \ldots \vee Ka_l \vee Kc_1 \vee \ldots \vee Kc_n \leftarrow Kb_1, \ldots, Kb_m, \tag{7}$$

for $a \in \Sigma$, respectively for every rule $r \in P$ of the form (1).

Then, the set of all semi-equilibrium models is given by $\{M \cap \Sigma^\kappa \mid M \in mc(AS(P^{HT}))\}$ and is denoted by $SEQ(P)$. In the following, we refer to semi-stable models or semi-equilibrium models as *paracoherent answer sets*, and we will be interested to consider only the true atoms of each paracoherent answer set. Hence, we denote by $SST^{\mathbf{t}}(P)$ the set $\{M \cap \Sigma \mid M \in SST(P)\}$, and by $SEQ^{\mathbf{t}}(P)$, the set $\{M \cap \Sigma \mid M \in SEQ(P)\}$.

3 Paracoherent Extensions

In this section, we formally introduce the paracoherent semantics for argumentation. Our goal is to provide another reasonable generalization of the stable semantics. Intuitively, we start identifying what is missing to a candidate set of arguments to become stable. This intuition is formalized by the basic concept of stabilizer.

Definition 3 (*Stabilizer*)
Let $F = (Ar, att)$ be an AF. A set of arguments $S \subseteq Ar$ is called a *stabilizer*, if there exists $A \subseteq Ar$, such that $A^+ \cup S^+ = Ar \setminus A$. We will say that S *is a stabilizer of* A, and also that A *admits* S *as stabilizer.*

Example 3
The AF reported in Figure 4, has the empty set as the unique admissible set of arguments. Now, let $A = \{a\}$. Since a attacks b and d, we have that $A^+ = \{b, d\}$. Therefore, a set of arguments S to be a stabilizer of A needs to attack at least arguments c and e, and is not allowed to attack a. So that, $S = \{b, d\}$ is a stabilizer of A. Indeed $A^+ \cup S^+ = \{b, c, d, e\} = Ar \setminus A$. Similarly, it can be seen that $\{a, b, d\}$, $\{b, d, e\}$ and $\{a, b, d, e\}$ are the remaining stabilizers of A.

Note that, in general, there are sets of arguments that do not admit any stabilizer. E.g., by considering the AF in Example 3, then the set $A' = \{a, b\}$ is such. This is due

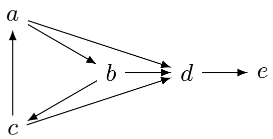


Fig. 4. Argumentation framework of the Examples 3, 4.

to the fact that a attacks b, thus $b \in A^+ \subseteq (A^+ \cup S^+)$, for each $S \subseteq Ar$. In general, we prove the following results.

Proposition 1
Let $F = (Ar, att)$ be an AF, and let $S, A \subseteq Ar$ be two sets of arguments. Then, (1) if S is a stabilizer of A, then A is conflict-free; (2) if S is a stabilizer of A, then $A \cap S^+ = \emptyset$; and (3) $\emptyset$ is a stabilizer of A iff A is stable.

To define argumentation semantics as close as possible to the stable extension, we need to select sets of arguments that admit stabilizers of minimal size. Thus, given an AF $F = (Ar, att)$, let $\Sigma_F = \{S \mid \exists A \subseteq Ar \text{ s.t. } S \text{ is a stabilizer of } A\}$ be the set of all possible stabilizers.

Definition 4 (*Paracoherent Extension*)
Let $F = (Ar, att)$ be an AF. A set of argument $A \subseteq Ar$ is a *paracoherent extension* of F, if A admits a minimal stabilizer among Σ_F w.r.t. $\subseteq$.

Example 4
Consider the AF of Example 3. Note that $\emptyset$ is not a stabilizer, since by Proposition 1 F does not admit any stable extension. However, for instance, $A = \{a, e\}$ is a paracoherent extension. Indeed, $S = \{b\}$ is a minimal stabilizer of A, as $(A^+ \cup S^+) = \{a, b, e\}^+ = \{b, c, d\} = Ar \setminus A$.

Now, we show formally that the introduced argumentation semantics behaves as an extension of the stable argumentation one such as the semi-stable and the stage semantics.

Theorem 1
Let $F = (Ar, att)$ be an AF. If $A \subseteq Ar$ is a stable extension, then A is also a paracoherent extension.

Proof
Let A be a stable extension. Then, by Proposition 1(3), $\emptyset$ is a stabilizer of A. As $\emptyset$ is minimal among all stabilizers in Σ_F , then A is a paracoherent extension. □

Theorem 2
Let $F = (Ar, att)$ be an AF. If $stb(F) \neq \emptyset$, then $para(F) = stb(F)$.

Proof
Assume that $stb(F) \neq \emptyset$. Hence, there is a stable extension A. Thus, by Theorem 1, A is a paracoherent extension. Now, let A' be a paracoherent extension. Since by assumption A is a stable extension, we know, by Proposition 1(3), that $\emptyset$ is a stabilizer of A. Hence, $\emptyset \in \Sigma_F$. Thus, $\emptyset$ is the unique minimal stabilizer. Therefore, it is also the minimal stabilizer of A'. Hence, again by Proposition 1(3), A' is a stable extension. □

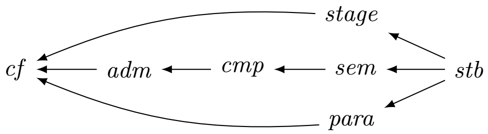


Fig. 5. Taxonomy of some argumentation semantics.

However, paracoherent semantics differs from both semi-stable semantics and stage semantics.

Example 5

Let F be the AF reported in Figure 3. In Example 2, we have seen that $sem(F) = \{\{a, d\}\}$, and $stage(F) = \{\{a, c, e\}, \{a, c, g\}, \{a, d, g\}\}$. Now, it can be checked that $para(F) = \{\{a, d\}, \{a, c, e\}, \{a, c, g\}, \{a, d, g\}\}$.

As it will be clearer in Section 6 paracoherent, semi-stable and stage are actually incomparable.

In Figure 5, a taxonomy of the argumentation semantics cited in the paper is reported. An arrow from a semantics σ to a semantics σ' means that $\sigma(F) \subseteq \sigma'(F)$, for each AF F, and there is an AF F' such that $\sigma(F) \subset \sigma'(F)$. Note that the paracoherent semantics is not admissible-based. E.g., the paracoherent extension $\{a, e\}$ of the Example 4 is not admissible, as the only admissible set of F is the empty set. Note that, the non-admissibility property is also shared by the *stage* semantics.

4 Relation with Logic Programming

In this section, we study the relation of our semantics for AFs with paracoherent semantics for logic programs. In particular, we show that paracoherent extensions of an AF F correspond to paracoherent answer sets of a logic program associated to F (Wu et al. 2009).

Definition 5

Let $F = (Ar, att)$ be an AF. For each argument $a \in Ar$, we build a rule r_a such that $H(r_a) = \{a\}$, $B^+(r_a) = \emptyset$, and $B^-(r_a) = \{c \in Ar \mid (c, a) \in att\}$. Then, we define P_F as the set $\{r_a \mid a \in Ar\}$.

Example 6

Consider the AF F reported in Figure 3. Its corresponding logic program is $P_{AF} = \{a;\ b \leftarrow not\ a;\ c \leftarrow not\ b,\ not\ d;\ d \leftarrow not\ c;\ e \leftarrow not\ d,\ not\ g;\ f \leftarrow not\ e,\ not\ f;\ g \leftarrow not\ f\}$.

Since there are two main paracoherent ASP semantics in logic programming, semi-stable model semantics and semi-equilibrium model semantics (see Preliminaries), we start to show that on logic programs having the particular structure of P_F, the two semantics coincide.

Theorem 3

Let P_F be the logic program corresponding to an AF F. Then, $SST^{\mathbf{t}}(P_F) = SEQ^{\mathbf{t}}(P_F)$.

Proof
First, the epistemic κ-transformation of P_F, P_F^κ, has a simplified structure. As each rule r in P_F is normal and it has an empty positive body (i.e. r is of the form $a \leftarrow \mathit{not}\ c_1, ..., \mathit{not}\ c_n$), then rule (2) becomes $\lambda_r \vee Kc_1 \vee ... \vee Kc_n$; rule (3) becomes $a \leftarrow \lambda_r$; rule (4) becomes $\leftarrow \lambda_r, c_j$; while rule (5) becomes irrelevant (it is $\lambda \leftarrow a, \lambda$), hence it can be removed. Moreover, as in P_F there is a unique rule having a in the head, then rules (2)-(3) can be unified in the rule $a \vee Kc_1 \vee ... \vee Kc_n$, and rule (4) is equivalent to $\leftarrow a, c_j$. Thus, the program P_F^κ is obtained by replacing each rule r, such that $B^-(r) \neq \emptyset$, with: (8) $a \vee Kc_1 \vee ... \vee Kc_n$ and (9) $\leftarrow a, c_j$, for $j = 1, ..., n$. Concerning the epistemic HT-transformation of P_F, P_F^{HT}, rule (7) becomes $Ka \vee Kc_1 \vee ... \vee Kc_n$. However, this rule can be derived by rule (8) and rule (6). Hence, it can be removed. Therefore, the program P_F^{HT} is obtained by adding to P_F^κ, only rules of the form (6), i.e., $Ka \leftarrow a$. Now, it is easy to see that if M is a minimal model of P_F^κ, then $M \cup \{Ka \mid a \in M \text{ and } Ka \notin M\}$ is a minimal model of P_F^{HT}; and if M' is a minimal model of P_F^{HT}, then there is $M \subseteq M'$, where eventually some Ka atoms are removed, such that M is a minimal model of P_F^κ. As both maps do not change the gap of the models and the true atoms, then $\{M \cap \Sigma \mid M \in mc(AS(P_F^\kappa))\} = \{M \cap \Sigma \mid M \in mc(AS(P_F^{HT}))\}$, i.e., $SST^{\mathbf{t}}(P_F) = SEQ^{\mathbf{t}}(P_F)$. □

Paracoherent extensions of an AF F coincide with paracoherent answer sets of P_F.

Theorem 4
Let P_F be the logic program corresponding to an AF F. Then, $\mathit{para}(F) = SEQ^{\mathbf{t}}(P_F)$.

Proof Sketch
Starting from $F = (Ar, att)$, we construct an AF $F_s = (Ar_s, att_s)$ as follows. For each $a \in Ar$, such that $(a, b) \in att$ for some $b \in Ar$, we consider a new argument sa, and a new attack (sa, b). So that, $Ar_s = Ar \cup \{sa \mid \exists b \in Ar, \text{ s.t. } (a, b) \in att\}$, and $att_s = att \cup \{(sa, b) \mid \exists b \in Ar, \text{ s.t. } (a, b) \in att\}$. Now, we compute the stable extensions of F_s. Then, we consider the minimal stable extensions with respect to the subset inclusion with respect to the arguments in $Ar_s \setminus Ar$, i.e., the set of all $A \in \mathit{stb}(F_s)$ for which there is no $A' \in \mathit{stb}(F_s)$ such that $A' \cap (Ar_s \setminus Ar) \subset A \cap (Ar_s \setminus Ar)$. We denote by $\mathit{mstb}(F_s)$ the set of all minimal stable extensions of F_s. Finally, by filtering the new atoms, we can prove that what we obtain is exactly the set of paracoherent extensions of F, i.e., $\mathit{para}(F) = \{A \cap Ar \mid A \in \mathit{mstb}(F_s)\}$.

To conclude the proof, we note that the stable extensions of F_s coincide with the answer sets of P_{F_s}, i.e., $\mathit{stb}(F_s) = AS(P_{F_s})$ (Dung 1995). Now, in (Amendola et al. 2018) has been introduced a semantical characterization for semi-stable models in terms of minimal externally supported (MES) models, by replacing the κ-transformation of P with the so-called externally supported program of P. That is, rules of the form (2)-(5) are replaced by rule $a_1 \vee \ldots \vee a_l \leftarrow b_1, \ldots, b_m, \mathit{not}\ c_1, \ldots, \mathit{not}\ c_n, \mathit{not}\ sc_1, \ldots, \mathit{not}\ sc_n$, and then choice rules of the form $\{sc_j\}$ are used to minimize the number of atoms of the form sc_j. Since we are dealing with logic programs of the form of P_F, the rule above becomes $a \leftarrow \mathit{not}\ c_1, \ldots, \mathit{not}\ c_n, \mathit{not}\ sc_1, \ldots, \mathit{not}\ sc_n$, that is exactly what the translation from F to F_s has done before. Therefore, it is easy to check that $MES(P_F)$ $= \{M \in AS(P_{F_s}) \mid \nexists M' \in AS(P_{F_s}) \text{ s.t. } s(M') \subset s(M)\} = \mathit{mstb}(F_s)$. Moreover, by Theorems 3-4 in (Amendola et al. 2018), we conclude that $SST^{\mathbf{t}}(P_F) = MES^{\mathbf{t}}(P_F)$, where $MES^{\mathbf{t}}(P_F) = \{M \cap \Sigma \mid M \in MES(P_F)\}$. Hence, $SST^{\mathbf{t}}(P_F) = \{A \cap Ar \mid A \in \mathit{mstb}(F_s)\} = \mathit{para}(F)$. □

Example 7
Consider again the AF F reported in Figure 3. The paracoherent answer sets of P_F (see Example 6) are $\{a, c, e, Kf\}$, $\{a, c, g, Ke\}$, $\{a, d, g, Ke\}$, and $\{a, d, Kf\}$. Hence, $\mathit{SEQ}^{\mathbf{t}}(P_F) = \{\{a, c, e\}, \{a, c, d\}, \{a, d, g\}, \{a, d\}\}$, that is equal to $\mathit{para}(F)$, as shown in the Example 5.

Note that, as it happens in the case of the relationship between the semi-stable and L-stable semantics (Caminada et al. 2015), if we encode a logic program P into an AF F_P using the association presented in (Caminada et al. 2015), it is not guaranteed that the paracoherent extensions of F_P can be mapped one to one with the paracoherent answer sets of P. E.g., if we consider the program $P = \{b \leftarrow \mathit{not}\ \ a;\ c \leftarrow b, \mathit{not}\ \ c\}$, the paracoherent answer sets of P are $\{Ka\}$ and $\{b, Kc\}$, whereas F_P has only one paracoherent extension corresponding to $\{b, Kc\}$.

5 Computational Complexity

In this section we study computational complexity issues. In particular, we prove that the credulous and skeptical reasoning tasks for the paracoherent argumentation semantics remain the same as for semi-stable and stage semantics.

Theorem 5
For paracoherent semantics, credulous reasoning is Σ_2^P-complete, and skeptical reasoning is Π_2^P-complete.

Proof Sketch
The memberships are a corollary of the Theorem 4. Indeed, credulous and skeptical reasoning for paracoherent semantics coincide with brave and cautious reasoning for semi-equilibrium model semantics, resp.; and it is well-known that for semi-equilibrium semantics on normal logic programs, credulous (brave) reasoning is Σ_2^P-complete, and skeptical (cautious) reasoning is Π_2^P-complete (Eiter et al. 2010). Concerning the hardness part, it can be checked that the proof of the hardness for credulous (brave) and skeptical (cautious) reasoning for semi-equilibrium model semantics in case of normal logic programs (see Theorem 10 and Appendix C1 in (Eiter et al. 2010)) can be directly used to prove that also for normal logic programs with an empty positive body and a single rule for each head atom (as it is P_F) hardness results remain unchanged. □

Finally, it is worthy to note that credulous and skeptical reasoning tasks associated with paracoherent argumentation semantics satisfy the criteria identified by (Gaggl et al. 2015), being at the second level of the polynomial hierarchy.

6 Related Work

In this section we review related works by first mentioning the relation with paracoherent semantics and, then, comparing in detail with alternative argumentation semantics.

6.1 Paracoherent semantics

The two major paracoherent semantics for logic programs are the semi-stable (Sakama and Inoue 1995) and the semi-equilibrium semantics (Amendola et al. 2016). These semantics emerged over several alternative proposals (Przymusinski 1991a; Gelder et al. 1991; Saccà and Zaniolo 1991; You and Yuan 1994; Eiter et al. 1997; Seipel 1997; Balduccini and Gelfond 2003; Pereira and Pinto 2005; 2007; Alcântara et al. 2005; Galindo et al. 2008). However, (Amendola et al. 2016) have shown that only semi-stable semantics (Sakama and Inoue 1995) and semi-equilibrium semantics (Amendola et al. 2016) satisfy all the following five highly desirable –from the knowledge representation point of view– theoretical properties: (i) every consistent answer set of a program corresponds to a paracoherent answer set (*answer set coverage*); (ii) if a program has some (consistent) answer set, then its paracoherent answer sets correspond to answer sets (*congruence*); (iii) if a program has a classical model, then it has a paracoherent answer set (*classical coherence*); (iv) a minimal set of atoms should be undefined (*minimal undefinedness*); (v) every true atom must be derived from the program (*justifiability or foundedness*). The first two properties ensure that the notions of answer sets and paracoherent answer sets should coincide for coherent programs; the third states that paracoherent answer sets should exist whenever the program admits a (classical) model; the last two state that the number of undefined atoms should be minimized, and every true atom should be derived from the program, respectively. The partial evidential stable models of (Seipel 1997) are known to be equivalent to semi-equilibrium ones (Amendola et al. 2016). An alternative characterization of semi-stable and the semi-equilibrium semantics based on the concept of *externally supported atoms* was given in (Amendola et al. 2018), that demonstrated to be amenable to obtain efficient implementations.

6.2 Comparison with Argumentation Semantics

In this section, we compare paracoherent semantics with alternative semantics for AFs. First, we compare paracoherent semantics with semi-stable semantics with respect to some basic features of admissible-based semantics. Then, we compare paracoherent semantics with stage semantics by focusing on the behaviour on unattacked arguments; and we give an extensive comparison with many argumentation semantics by focusing on a specific desirable behaviour that paracoherent semantics exhibits, and other semantics do not. Finally, we provide a discussion concerning the connection with some other studies on the existence of stable extensions.

6.2.1 Cycles and non-admissibility: Paracoherent Semantics vs Semi-stable Semantics

Great attention to the problem of loops and cycles has been given both in argumentation (Bench-Capon 2016; Gabbay 2016; Dvorák and Gaggl 2016; Arieli 2016; Bodanza et al. 2016), and in answer set programming (Lee and Lifschitz 2003; Lin and Zhao 2004; Costantini and Provetti 2005; Gebser et al. 2011). An AF described by an odd-length cycle admits the empty admissible set only. Hence, admissible-based semantics, such as complete, grounded, preferred, semi-stable, stable, ideal (Dung et al. 2007), eager (Caminada 2007), and resolution-based grounded (Baroni et al. 2011) can (eventually) admit the empty set as the unique solution. This is the case, in particular, for semi-stable semantics. However, in case of AFs described by even-length cycles several extended solution are possible. For instance, if $F = (\{a, b, c, d\}, \{(a, b), (b, c), (c, d), (d, a)\})$ (the 4-length cycle), we have two stable extensions, $\{a, c\}$ and $\{b, d\}$, that are also semi-stable extensions.

The paracoherent semantics is non-admissible, unlike semi-stable. In many situations, non-admissible semantics allow one to have solutions that are not empty. This is a desirable behavior in practice as noted in (Verheij 1996). Indeed, it is known that a small initial incoherence might prevents to draw interesting conclusions with the semi-stable semantics. Intuitively, this happens when the argumentation framework "starts" with an odd-cycle (i.e., an odd-cycle appears in a strongly connected component of the graph and no argument of this component is attacked by others).

For example, consider the following argumentation framework:

$$F = (\{a, b, c, d, e\}, \{(a, a), (a, b), (b, c), (c, d), (d, e)\}).$$

The starting loop on the node a avoids to have a non-empty admissible solution. Hence, the only semi-stable extension is the empty set. On the other hand, the paracoherent approach allows to obtain solutions also in case of "initial incoherence". For instance, in this example, we have $\{c, e\}$ as (the unique) paracoherent solution.

The paracoherent semantics, like others non admissible-based semantics (such as stage), guarantees the relevant feature of ensuring a "symmetric" treatment of odd- and even-length cycles (Baroni et al. 2005; Baroni and Giacomin 2007; Dvorák and Gaggl 2016). A symmetric behavior is strictly connected with the admissibility property, that guarantees to each argument in a conflict-free solution to be defended by another argument. This cannot happen when we are in the presence of odd cycles. Hence, the unique solution must be the empty one, i.e., we suspend any judgment in the presence of inconsistencies (as in semi-stable). However, in real scenarios this might not suffice. Consider for example a person charged with a crime. In practice, it will be punished or not. A decision will however be made. In such circumstances, we need to keep the argumentation system capable of providing plausible solutions. For these reasons, we believe that it is necessary to resort to non-admissible semantics. However, to minimize "non-admissibility" it is necessary to keep "as close as possible" to admissible solutions, and, in particular, to maintain a similar/symmetrical behavior to that of stable solutions.

It is not a case that non-admissible semantics can be considered a natural way of evaluating argumentation frameworks. This was evidenced by a recent empirical study concluding that non-admissible semantics "were the best predictors of human argument evaluation" (Cramer and Guillaume 2019).

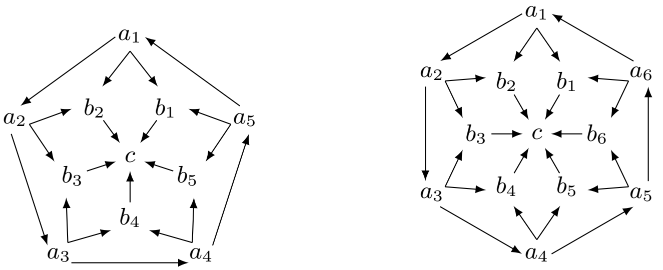
6.2.2 Unattacked arguments: Paracoherent Semantics vs Stage Semantics

Paracoherent semantics provides a better behaviour in case of "unattacked arguments" (see, (Caminada et al. 2012)) when compared with the stage semantics.

For example, consider the following argumentation framework:

$$F = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, c), (c, d), (d, e)\}),$$

where argument a attacks b; b attacks c; c attacks itself and d; and d attacks e. Note that, in particular, a is attacked by no argument. The argumentation framework F has two stage extensions, namely $A_1 = \{a, d\}$ and $A_2 = \{b, d\}$. Indeed, $A_1 \cup A_1^+ = \{a, b, d, e\}$ and $A_2 \cup A_2^+ = \{b, c, d, e\}$, and it can be easily checked that they are maximal with respect to conflict-free sets.



(a) 5-radial star polygon

(b) 6-radial star polygon

Fig. 6. A case of symmetric attacks.

Now, intuitively, it is very strange that the argument a is false in some extension, namely A_2, because a is an argument attacked by no other argument. There is no reason to consider a false. Hence, one expects to see a as true in each extension, that is a should be a skeptical argument.

This expected behaviour is maintained by paracoherent extensions. Indeed, it can be checked that $B_1 = \{a, d\}$ and $B_2 = \{a, e\}$ are all the paracoherent extensions. Thus, a is a skeptical argument as expected.

6.2.3 A distinguishing feature of Paracoherent Semantics: The Symmetric Behaviour

We now provide an extensive comparison between paracoherent semantics and several argumentation semantics. We discuss a distinguishing feature of the paracoherent semantics: its symmetric behaviour.

Consider the AF $F_n = (Ar, att)$, where $Ar = \{a_1, ..., a_n, b_1, ..., b_n, c\}$, and att is formed by the cycle of length n given by (a_i, a_{i+1}), for $i = 1, ..., n-1$, and (a_n, a_1); the pairs of attacks (a_i, b_i), (a_i, b_{i+1}), for $i = 1, ..., n-1$, and (a_n, b_n), (a_n, b_1); and finally (b_i, c), for $i = 1, ..., n$. Intuitively, a_1, ..., a_n form a cycle of attacks; each a_i attacks two arguments subscripted as consecutive, b_i and b_{i+1}, except for the last a_n which attacks b_n and b_1, by completing a sort of cyclic attack; and each b_i attacks the argument c. We will call this graph the *n-radial star polygon*. Figure 6 reports what is happening in case of $n = 5$ and $n = 6$.

In case of n is even, we obtain two stable extensions: $\{a_1, a_3, ..., a_{n-1}, c\}$ and $\{a_2, a_4, ..., a_n, c\}$. Note that c, in particular, is always inferred, so it is a skeptical argument. However, as well-known, in case of n is odd, the unique admissible set is the empty one, whereas no stable extension exists. We stress that no admissible-based semantics can have a symmetric behaviour with respect to cycles. However, also non admissible-based semantics have this issue with cycles. This happens, for instance, for $cf2$ (Baroni et al. 2005; Gaggl and Woltran 2013) argumentation semantics, as shown by (Dvorák and Gaggl 2016). In our symmetric example, for $cf2$, the extensions of F_5 are $\{a_1, a_3, b_5\}$, $\{a_2, a_4, b_1\}$, $\{a_3, a_5, b_2\}$, $\{a_4, a_1, b_3\}$, and $\{a_5, a_2, b_4\}$, thus c does not belong to any of them. Nonetheless, among the $cf2$ extensions of F_6 there are $\{a_1, a_3, a_5, c\}$, $\{a_2, a_4, a_6, c\}$, $\{a_1, a_4, b_3, b_6\}$, $\{a_2, a_5, b_4, b_1\}$, $\{a_3, a_6, b_5, b_2\}$, hence c is not skeptically but credulously accepted, thus resulting in a different behavior for even and odd radial star polygons.

This non-symmetric behaviour happens also for the stage semantics. Since the stage semantics is a generalization of the stable semantics, and, whenever a stable extension exists, c is a skeptical argument, then, to fulfill a symmetric behaviour, the stage semantics should guarantee the skeptical acceptance of c. However, just considering $n = 3$, the stage extensions will be $\{a_1, b_3\}$, $\{a_2, b_1\}$, and $\{a_3, b_2\}$. Thus, c is not even a credulous argument. Note that this holds also for extensions of stage semantics such as the *stagle* (Baumann et al. 2016).

Therefore, admissible, complete, grounded, ideal, preferred, stable, semi-stable, eager, resolution-based grounded, stage, and cf2 semantics do not have a symmetric behaviour w.r.t. radial star polygons. While, paracoherent semantics on n-radial star polygon infers c as skeptical argument, for each n. In particular, whenever n is odd, the set of the paracoherent extensions of F_n is formed by $A_i = \{c\} \cup \{a_{i-2h \pmod n} : h = 0, 1, ..., \frac{n-3}{2}\}$, for each $i = 1, ..., n$.

6.2.4 Further studies on the non-existence of stable extensions

As a final observation, we note that the reasons for non-existence of stable extensions has been recently investigated in (Schulz and Toni 2018). In particular, the idea of (Schulz and Toni 2018) is to "fix" preferred extensions that are not stable by applying a structural revision of the original AF. This revision can be used to make stable some preferred extensions, no matter whether the AF admits a stable one. This goal is rather different from the ideas underlying the paracoherent semantics which aims at finding a (minimal) remedy to missing stable extensions without modifying the original AF. Nonetheless, an interesting open question is whether stabilizers are in some way related to the concept of responsible sets of (Schulz and Toni 2018).

7 Conclusion

This paper introduces a different perspective on AFs with no stable extension, by proposing the paracoherent extensions. The new semantics coincides with the stable semantics, whenever a stable extension exists, and has a natural counterpart in paracoherent semantics for logic programs. Moreover, we studied the computational complexity of the main reasoning tasks, that remain unchanged in comparison with semi-stable and stage semantics. Finally, we compared paracoherent semantics with several existing argumentation semantics, by showing an interesting distinctive symmetric behaviour on graphs that involve odd-length cycles. In the literature of Argumentation frameworks several semantics have been proposed, each one having some distinctive feature. It is difficult, and probably impossible, to identify an overall winner in this context, where often proposals are incomparable. However, it can be observed that symmetry is very common property in nature, and has been often subject of cross-disciplinary studies. Already our running example demonstrates the desirable symmetric behaviour of our semantics is useful while modelling a very well known problem (Stable Roommates) with a large number of applications in real world (Gale and Shapley 1962). Our contribution is also relevant because it adds a missing link in the panorama of correspondences of AF semantics with logic programming ones.

As future work, we will deepen the study of the symmetrical behavior and to study additional properties of our semantics as done in (Baroni and Giacomin 2007). Moreover, we plan to implement paracoherent semantics to solve typical reasoning problems considered in competitions (Gaggl et al. 2016). A possibility would be to exploit the techniques developed in (Amendola et al. 2017; Amendola et al. 2019a; Amendola et al. 2019b) and devise an evaluation procedure that works by rewriting AFs in ASP programs.

Acknowledgements

This work has been partially supported by MIUR under PRIN 2017 project n. 2017$M9C25L$_001.

References

Alcântara, J., Damásio, C. V., and Pereira, L. M. 2005. An encompassing framework for paraconsistent logic programs. *J. Applied Logic 3,* 1, 67–95.

Amendola, G. 2018. Solving the stable roommates problem using incoherent answer set programs. In *RiCeRcA@AI*IA*. CEUR Workshop Proceedings, vol. 2272. CEUR-WS.org.

Amendola, G., Dodaro, C., Faber, W., Leone, N., and Ricca, F. 2017. On the computation of paracoherent answer sets. In *AAAI*. AAAI Press, 1034–1040.

Amendola, G., Dodaro, C., Faber, W., Pulina, L., and Ricca, F. 2019. Algorithm selection for paracoherent answer set computation. In *JELIA*. Lecture Notes in Computer Science, vol. 11468. Springer, 479–489.

Amendola, G., Dodaro, C., Faber, W., and Ricca, F. 2018. Externally supported models for efficient computation of paracoherent answer sets. In *AAAI*. AAAI Press, 1720–1727.

Amendola, G., Dodaro, C., and Ricca, F. 2019. Better paracoherent answer sets with less resources. *TPLP this volume,* this number, to appear.

Amendola, G., Eiter, T., Fink, M., Leone, N., and Moura, J. 2016. Semi-equilibrium models for paracoherent answer set programs. *Artif. Intell. 234*, 219–271.

Amgoud, L., Parsons, S., and Maudet, N. 2000. Arguments, dialogue, and negotiation. In *ECAI*. IOS Press, 338–342.

Arieli, O. 2016. On the acceptance of loops in argumentation frameworks. *J. Log. Comput. 26,* 4, 1203–1234.

Balduccini, M. and Gelfond, M. 2003. Logic programs with consistency-restoring rules. In *ISLFCR, AAAI 2003 Spring Symposium Series*. 9–18.

Baroni, P., Dunne, P. E., and Giacomin, M. 2011. On the resolution-based family of abstract argumentation semantics and its grounded instance. *Artif. Intell. 175,* 3-4, 791–813.

Baroni, P., Gabbay, D. M., and Giacomin, M. 2016. Introduction to the special issue on loops in argumentation. *J. Log. Comput. 26,* 4, 1051–1053.

Baroni, P. and Giacomin, M. 2007. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell. 171,* 10-15, 675–700.

Baroni, P., Giacomin, M., and Guida, G. 2005. Scc-recursiveness: a general schema for argumentation semantics. *Artif. Intell. 168,* 1-2, 162–210.

Baumann, R., Linsbichler, T., and Woltran, S. 2016. Verifiability of argumentation semantics. In *COMMA'16*. 83–94.

Bench-Capon, T. J. M. 2016. Dilemmas and paradoxes: cycles in argumentation frameworks. *J. Log. Comput. 26,* 4, 1055–1064.

Bench-Capon, T. J. M. and Dunne, P. E. 2007. Argumentation in artificial intelligence. *Artif. Intell. 171,* 10-15, 619–641.

Bodanza, G. A., Tohmé, F. A., and Simari, G. R. 2016. Beyond admissibility: accepting cycles in argumentation with game protocols for cogency criteria. *J. Log. Comput. 26,* 4, 1235–1255.

Brewka, G., Eiter, T., and Truszczynski, M. 2011. Answer set programming at a glance. *Commun. ACM 54,* 12, 92–103.

Caminada, M. 2006. Semi-stable semantics. In *COMMA'06*. 121–130.

Caminada, M. 2007. Comparing two unique extension semantics for formal argumentation: ideal and eager. In *BNAIC'07*.

Caminada, M., Sá, S., Alcântara, J., and Dvorák, W. 2015. On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reasoning 58*, 87–111.

Caminada, M. W. A., Carnielli, W. A., and Dunne, P. E. 2012. Semi-stable semantics. *J. Log. Comput. 22,* 5, 1207–1254.

Costantini, S. and Provetti, A. 2005. Normal forms for answer sets programming. *TPLP 5,* 6, 747–760.

Cramer, M. and Guillaume, M. 2019. Empirical study on human evaluation of complex argumentation frameworks. In *JELIA*. Lecture Notes in Computer Science, vol. 11468. Springer, 102–115.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell. 77,* 2, 321–358.

Dung, P. M., Mancarella, P., and Toni, F. 2007. Computing ideal sceptical argumentation. *Artif. Intell. 171,* 10-15, 642–674.

Dunne, P. E. and Caminada, M. 2008. Computational complexity of semi-stable semantics in abstract argumentation frameworks. In *JELIA'08*. 153–165.

Dunne, P. E. and Wooldridge, M. 2009. Complexity of abstract argumentation. In *Argumentation in Artificial Intelligence*. 85–104.

Dvorák, W. and Gaggl, S. A. 2016. Stage semantics and the scc-recursive schema for argumentation semantics. *J. Log. Comput. 26,* 4, 1149–1202.

Dvorák, W. and Woltran, S. 2010. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Process. Lett. 110,* 11, 425–430.

Eiter, T., Fink, M., and Moura, J. 2010. Paracoherent answer set programming. In *KR*. AAAI Press.

Eiter, T., Leone, N., and Saccà, D. 1997. On the partial semantics for disjunctive deductive databases. *Ann. Math. Artif. Intell. 19,* 1-2, 59–96.

Gabbay, D. M. 2016. The handling of loops in argumentation networks. *J. Log. Comput. 26,* 4, 1065–1147.

Gaggl, S. A., Linsbichler, T., Maratea, M., and Woltran, S. 2016. Introducing the second international competition on computational models of argumentation. In *SAFA@COMMA*. 4–9.

Gaggl, S. A., Linsbichler, T., Maratea, M., and Woltran, S. 2018. Summary report of the second international competition on computational models of argumentation. *39,* 4, 77–79.

Gaggl, S. A., Rudolph, S., and Thomazo, M. 2015. What is a reasonable argumentation semantics? In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*. 309–324.

Gaggl, S. A. and Woltran, S. 2013. The cf2 argumentation semantics revisited. *J. Log. Comput. 23,* 5, 925–949.

Gale, D. and Shapley, L. S. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly 69*, 9–15.

Galindo, M. J. O., Ramírez, J. R. A., and Carballido, J. L. 2008. Logical weak completions of paraconsistent logics. *J. Log. Comput. 18,* 6, 913–940.

Gebser, M., Lee, J., and Lierler, Y. 2011. On elementary loops of logic programs. *TPLP 11,* 6, 953–988.

Gelder, A. V., Ross, K. A., and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *J. ACM 38,* 3, 620–650.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, August 15-19, 1988 (2 Volumes).* 1070–1080.

Gelfond, M. and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput. 9,* 3/4, 365–386.

Jakobovits, H. and Vermeir, D. 1999. Robust semantics for argumentation frameworks. *J. Log. Comput. 9,* 2, 215–261.

Lee, J. and Lifschitz, V. 2003. Loop formulas for disjunctive logic programs. In *ICLP.* Lecture Notes in Computer Science, vol. 2916. Springer, 451–465.

Lifschitz, V. 1999. Answer set planning. In *ICLP.* MIT Press, 23–37.

Lin, F. and Zhao, X. 2004. On odd and even cycles in normal logic programs. In *AAAI.* AAAI Press / The MIT Press, 80–85.

Pereira, L. M. and Pinto, A. M. 2005. Revised stable models - a semantics for logic programs. In *EPIA.* 29–42.

Pereira, L. M. and Pinto, A. M. 2007. Approved models for normal logic programs. In *LPAR.* 454–468.

Pollock, J. L. 1998. The logical foundations of goal-regression planning in autonomous agents. *Artif. Intell. 106,* 2, 267–334.

Prakken, H. and Sartor, G. 1996. A dialectical model of assessing conflicting arguments in legal reasoning. *Artif. Intell. Law 4,* 3-4, 331–368.

Przymusinski, T. C. 1991a. Stable semantics for disjunctive programs. *New Generation Comput. 9,* 3/4, 401–424.

Przymusinski, T. C. 1991b. Three-valued nonmonotonic formalisms and semantics of logic programs. *Artif. Intell. 49,* 1-3, 309–343.

Saccà, D. and Zaniolo, C. 1991. Partial models and three-valued models in logic programs with negation. In *LPNMR.* The MIT Press, 87–101.

Sakama, C. and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *J. Log. Comput. 5,* 3, 265–285.

Schulz, C. and Toni, F. 2018. On the responsibility for undecisiveness in preferred and stable labellings in abstract argumentation. *Artif. Intell. 262*, 301–335.

Seipel, D. 1997. Partial evidential stable models for disjunctive deductive databases. In *LPKR.* Lecture Notes in Computer Science, vol. 1471. Springer, 66–84.

Simari, G. R. and Rahwan, I., Eds. 2009. *Argumentation in Artificial Intelligence.* Springer.

Simons, P., Niemelä, I., and Soininen, T. 2002. Extending and implementing the stable model semantics. *Artif. Intell. 138,* 1-2, 181–234.

Strass, H. 2013. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell. 205*, 39–70.

Verheij, B. 1996. Two approaches to dialectical argumentation: Admissible sets and argumentation stages. In *FAPR.* 357–368.

Verheij, B. 2003. Artificial argument assistants for defeasible argumentation. *Artif. Intell. 150,* 1-2, 291–324.

Wu, Y., Caminada, M., and Gabbay, D. M. 2009. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica 93,* 2-3, 383–403.

You, J. and Yuan, L. 1994. A three-valued semantics for deductive databases and logic programs. *J. Comput. Syst. Sci. 49,* 2, 334–361.