Journal of Logic and Computation Advance Access published February 13, 2014

On the acceptance of loops in argumentation frameworks

OFER ARIELI, *School of Computer Science, The Academic College of Tel-Aviv, 2 Rabenu Yerucham Street, P.O.Box 8401, Tel-Aviv 61083, Israel. E-mail: oarieli@mta.ac.il*

Abstract

Current approaches for giving semantics to abstract argumentation frameworks dismiss altogether any possibility of having conflicts among accepted arguments by requiring that the latter should be 'conflict free'. In reality, however, contradictory phenomena coexist, or it may happen that one cannot make a choice between conflicting indications but still would like to keep track to all of them. For this purpose we introduce in this article a new kind of argumentation semantics, called 'conflict-tolerant', in which all the accepted arguments must be justified (in the sense that each one of them can be defended), but some of them may still attack each other. In terms of graphical representation of argumentation systems, where attacks are represented by directed edges, this means that the possibility of accepting 'loops' of arguments is not automatically ruled out without any further considerations. To provide conflict-tolerant semantics, we enhance the two standard approaches for defining coherent (conflict-free) semantics for argumentation frameworks. The extension-based approach is generalized by relaxing the 'conflict-freeness' requirement of the chosen sets of arguments, and the three-valued labelling approach is replaced by a four-valued labelling system that allows to capture mutual attacks among accepted arguments. We show that our setting is not a substitute of standard (conflict-free) semantics, but rather a generalized framework that accommodates both conflict-free and conflict-tolerant semantics. Moreover, the one-to-one relationship between extensions and labellings of conflict-free semantics is carried on to a similar correspondence between the extended approaches for providing conflict-tolerant semantics. Thus, in our setting as well, these are essentially two points of views for the same thing.

Keywords: Abstract argumentation, conflict-tolerant semantics, 4-valued labeling, paraconsistent extensions.

1 Introduction and motivation

An abstract argumentation framework consists of a set of (abstract) arguments and a binary relation that intuitively represents attacks between arguments [22]. A semantics for such a structure is an indication which arguments can collectively be accepted in a rational way in light of the attack relation. A starting point of all the existing semantics for abstract argumentation frameworks is that their set(s) of acceptable arguments must be *conflict-free*, i.e., an accepted argument should not be attacked by another accepted argument. This means, in particular, a dismissal of any self-referring argument and a rejection of any contradictory set of arguments. However, in everyday life it is not always the case that theories are completely coherent even when each of their arguments provides a solid and acceptable assertion, and so contradictory sets of arguments should sometimes be recognized and tolerated. Moreover, a removal of contradictory indications in such theories may imply a loss of information and may lead to erroneous conclusions. This is exemplified next.

EXAMPLE 1.1
The phenomena of interference on one hand and the photoelectric effect on the other hand may stand behind conflicting arguments about whether light is a particle or a wave. Any choice between such arguments would obviously be arbitrary, and the dismissal of one of them would unavoidably yield erroneous conclusions about the nature of light. For having a realistic theory, it is therefore essential in this case to adopt an attitude that tolerates both conflicting arguments.

doi:10.1093/logcom/exu009

Another situation where conflicting arguments may be accepted is when a gullible approach is beneficial. This is demonstrated by the next example.

Example 1.2
The following is a variation of the decision-making problem presented in [28]: suppose that a traveler has doubts whether to take a coat or sunglasses to her journey. She consults with two weather websites, one says that the weather in her destination is rainy, while the other one says that the weather is sunny. If one website is considered more reliable than the other, the traveler may act accordingly. However, if the web-sources are equally reliable, the traveler still has *two* options for making a choice: she may withhold any action and wait until the weather conditions are clarified, or she may take a more practical decision and take both a coat and sunglasses. The later is a pragmatic approach, accepting contradictory indications whenever this does not cause any real risk or damage. In other situations, for instance when there are conflicting symptoms obliging different medical treatments, it may be more rational to refrain from irreversible acts. In both cases, though, the two neutral options have totally different consequences, so it is useful to clearly distinguish between them (as we do in what follows).

In this article, we consider a new approach for argumentation semantics, accommodating conflicting arguments and making a clear distinction between two kinds of uncertainty in argumentation: insufficient or irrelevant arguments on one hand and conflicting or ambiguous arguments on the other hand. For this, we extend the following standard approaches of defining semantics to abstract argumentation frameworks:

- *Extension-based semantics.* According to this method, the semantics of a given argumentation framework (i.e., the consequences of a dispute) is determined by sets of arguments (called extensions) that can collectively be accepted. According to standard extension-based approaches, all the extensions must have at least two primary properties: admissibility and conflict-freeness (see, e.g. [9, 11]). The former property, guaranteeing that an extension Ext ‘defends’ all of its elements (i.e. Ext ‘counterattacks’ each argument that attacks some $e \in Ext$), is preserved also in our framework, since otherwise acceptance of arguments would be an arbitrary choice. However, the other property is lifted in our case, since—as indicated above—we would like to permit, in some situations, conflicting arguments.
- *Labeling-based semantics.* According to this method, each argument is assigned a label that designates its status (accepted, rejected, undecided—see [17, 19]). We extend this traditional three-states labelings of arguments by a fourth state, so now apart from accepting or rejecting an argument we have *two* additional states, representing two opposite reasons for avoiding a definite opinion about the argument as hand: one (‘none’), indicating that there is too little evidence for reaching a precise conclusion about the argument’s validity, and the other (‘both’) indicating ‘too much’ (contradictory) evidence, i.e. the existence of both supportive and opposing arguments concerning the argument under consideration.

Both of the generalized approaches described above are a conservative extension of the standard approaches for giving semantics to abstract argumentation systems, in the sense that they do not exclude standard extensions or labelings, but rather offer additional points of views to the state of affairs as depicted by the argumentation framework. This allows us to introduce a brand new family of semantics that tolerate conflicts in the sense that internal attacks among accepted arguments are allowed, while the set of accepted arguments is not trivialized (i.e. it is not the case that every argument is necessarily accepted).

We introduce an extended set of criteria for selecting the most plausible four-valued labelings for an argumentation framework. These criteria are then justified by showing that the one-to-one relationship between extensions and labelings obtained for conflict-free semantics (see [19]) is carried onto a similar correspondence between the extended approaches for providing conflict-tolerant (paraconsistent) semantics. This also shows that in the case of conflict-tolerant semantics as well, extensions and labelings are each other's dual.

The rest of this article is organized as follows: In the next section, we briefly review some basic concepts and definitions behind abstract argumentation theory, including the two standard approaches mentioned above for giving semantics to argumentation frameworks. In Section 3, we introduce conflict-tolerant semantics for argumentation frameworks and show how the two standard semantic approaches can be generalized for accommodating conflicts in the accepted sets of arguments. Section 4 shows the correspondence between conflict-tolerant semantics and their conflict-free counterparts, and in Section 5 we demonstrate the usefulness of conflict-tolerant semantics in the context of constrained argumentation frameworks. Section 6 shows how conflict-tolerant semantics can be formalized in terms of propositional theories, and in Section 7 we conclude.

This article combines and extends the conference papers [4] (covered in Sections 2–4) and [5] (covered in Section 5 and part of Section 6).

2 Preliminaries

Let us first recall some basic definitions and useful notions regarding abstract argumentation theory.

Definition 2.1
A (finite) *argumentation framework* [22] is a pair $\mathcal{AF} = \langle Args, Att \rangle$, where $Args$ is a finite set, the elements of which are called *arguments*, and Att is a relation on $Args \times Args$ whose instances are called *attacks*. When $(A,B) \in Att$ we say that A *attacks* B (or that B is *attacked by* A).

Given an argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$, in the sequel we shall use the following notations for an argument $A \in Args$ and a set of arguments $\mathcal{S} \subseteq Args$:

- The set of arguments that are *attacked by* A is denoted A^+, i.e., $A^+ = \{B \in Args \mid (A,B) \in Att\}$.
- The set of arguments that *attack* A is denoted A^-, i.e., $A^- = \{B \in Args \mid (B,A) \in Att\}$.

Similarly, $\mathcal{S}^+ = \bigcup_{A \in \mathcal{S}} A^+$ and $\mathcal{S}^- = \bigcup_{A \in \mathcal{S}} A^-$ denote, respectively, the set of arguments that are attacked by some argument in $\mathcal{S}$ and the set of arguments that attack some argument in $\mathcal{S}$. Accordingly, we denote:

- The set of arguments that are *defended by* $\mathcal{S}$ is $\mathrm{Def}(\mathcal{S}) = \{A \in Args \mid A^- \subseteq \mathcal{S}^+\}$.

Thus, an argument A is defended by $\mathcal{S}$ if each attacker of A is attacked by (an argument in) $\mathcal{S}$. The two primary principles of acceptable sets of arguments are now defined as follows:

Definition 2.2
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework.

– A set $\mathcal{S} \subseteq Args$ is *conflict-free* (with respect to $\mathcal{AF}$) iff $\mathcal{S} \cap \mathcal{S}^+ = \emptyset$.
– A conflict-free set $\mathcal{S} \subseteq Args$ is *admissible* for $\mathcal{AF}$, iff $\mathcal{S} \subseteq \mathrm{Def}(\mathcal{S})$.

Conflict-freeness assures that no argument in the set is attacked by another argument in the set, and admissibility guarantees, in addition, that the set is self defendant. A stronger notion is the following:

– A conflict-free set $\mathcal{S} \subseteq Args$ is *complete* for $\mathcal{AF}$, iff $\mathcal{S} = \mathrm{Def}(\mathcal{S})$.

The principles defined above are a cornerstone of a variety of extension-based semantics, which formalize what sets of arguments can collectively be accepted from a given argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$ (see, e.g. [22, 29]). In what follows, we shall usually denote an extension by *Ext*. This includes, among others, *grounded extensions* (the minimal subset of *Args*, with respect to set inclusion, that is complete for $\mathcal{AF}$), *preferred extensions* (maximal subsets of *Args* that are complete for $\mathcal{AF}$), *stable extensions* (any complete subset *Ext* of *Args* for which $Ext^+ = Args \setminus Ext$) and so forth.[1]

An alternative way to describe argumentation semantics is based on the concept of an *argument labeling* [17, 19]. The main definitions and the relevant results concerning this approach are surveyed below.

Definition 2.3
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. An *argument labeling* is a complete function $lab : Args \to \{\mathsf{in}, \mathsf{out}, \mathsf{undec}\}$. We shall sometimes write $\mathsf{In}(lab)$ for $\{A \in Args \mid lab(A) = \mathsf{in}\}$, $\mathsf{Out}(lab)$ for $\{A \in Args \mid lab(A) = \mathsf{out}\}$ and $\mathsf{Undec}(lab)$ for $\{A \in Args \mid lab(A) = \mathsf{undec}\}$.

In essence, an argument labeling expresses a position on which arguments one accepts (labeled in), which arguments one rejects (labeled out) and which arguments one abstains from having an explicit opinion about (labeled undec). Since a labeling *lab* of $\mathcal{AF} = \langle Args, Att \rangle$ can be seen as a partition of *Args*, we shall sometimes write it as a triple $\langle \mathsf{In}(lab), \mathsf{Out}(lab), \mathsf{Undec}(lab) \rangle$.

Definition 2.4
Consider the following conditions on a labeling *lab* and an argument *A* in a framework $\mathcal{AF} = \langle Args, Att \rangle$:

Pos1	If $lab(A) = \mathsf{in}$ then there is no $B \in A^-$ such that $lab(B) = \mathsf{in}$.
Pos2	If $lab(A) = \mathsf{in}$ then for every $B \in A^-$ it holds that $lab(B) = \mathsf{out}$.
Neg	If $lab(A) = \mathsf{out}$ then there exists some $B \in A^-$ such that $lab(B) = \mathsf{in}$.
Neither	If $lab(A) = \mathsf{undec}$ then not for every $B \in A^-$ it holds that $lab(B) = \mathsf{out}$ and there does not exist a $B \in A^-$ such that $lab(B) = \mathsf{in}$.

Now, given a labeling *lab* of an argumentation framework $\langle Args, Att \rangle$, we say that

- *lab* is *conflict-free* (for $\mathcal{AF}$), if for every $A \in Args$ it satisfies conditions **Pos1** and **Neg**,
- *lab* is *admissible* (for $\mathcal{AF}$), if for every $A \in Args$ it satisfies conditions **Pos2** and **Neg**,
- *lab* is *complete* (for $\mathcal{AF}$), if for every $A \in Args$ it satisfies conditions **Pos2**, **Neg**, and **Neither**.[2]

Again, the labelings considered above serve as a basis for a variety of labeling-based semantics that have been proposed for an argumentation framework $\mathcal{AF}$, each one of which is a counterpart of a corresponding extension-based semantics. This includes, for instance, the *grounded labeling* (a complete labeling for $\mathcal{AF}$ with a minimal set of in-assignments), *preferred labelings* (complete labelings for $\mathcal{AF}$ with a maximal set of in-assignments), *stable labelings* (any complete labeling of $\mathcal{AF}$ without undec-assignments) and so forth.

[1] Common definitions of conflict-free extension-based semantics for argumentation frameworks, different methods for computing them and computational complexity analysis appear, e.g. in [7, 18, 22–25].

[2] In particular, a complete labeling is admissible.

The following correspondence between extensions and labelings is shown in [19]:[3]

PROPOSITION 2.5
Let $\mathcal{AF}=\langle Args, Att\rangle$ be an argumentation framework, $\mathcal{CFE}$ the set of all conflict-free extensions of $\mathcal{AF}$, and $\mathcal{CFL}$ the set of all conflict-free labelings of $\mathcal{AF}$. Consider the function $\mathcal{LE}_{\mathcal{AF}}:\mathcal{CFL}\rightarrow\mathcal{CFE}$, defined by $\mathcal{LE}_{\mathcal{AF}}(lab)=\mathsf{In}(lab)$ and the function $\mathcal{EL}_{\mathcal{AF}}:\mathcal{CFE}\rightarrow\mathcal{CFL}$, defined by $\mathcal{EL}_{\mathcal{AF}}(Ext)=\langle Ext, Ext^{+}, Args\setminus(Ext\cup Ext^{+})\rangle$. It holds that:

(1) If *Ext* is an admissible (respectively, complete) extension, then $\mathcal{EL}_{\mathcal{AF}}(Ext)$ is an admissible (respectively, complete) labeling.
(2) If *lab* is an admissible (respectively, complete) labeling, then $\mathcal{LE}_{\mathcal{AF}}(lab)$ is an admissible (respectively, complete) extension.
(3) When the domain and range of $\mathcal{EL}_{\mathcal{AF}}$ and $\mathcal{LE}_{\mathcal{AF}}$ are restricted to complete extensions and complete labelings of $\mathcal{AF}$, these functions become bijections and each other's inverses, making complete extensions and complete labelings one-to-one related.

3 Tolerance of conflicts

In this section, we extend the two approaches considered previously in order to define conflict-tolerant semantics for abstract argumentation frameworks. Recall that our purpose here is twofold:

(1) Introducing self-referring argumentation and avoiding information loss that may be caused by the conflict-freeness requirement (Thus, for instance, it may be better to accept extensions with a small fragment of conflicting arguments than, say, sticking to the empty extension).
(2) Refining the undec-indication in standard labeling systems, which reflects (at least) two totally different situations: one case is that the reasoner abstains from having an opinion about an argument because there are no indications whether this argument should be accepted or rejected. Another case that may cause a neutral opinion is that there are simultaneous considerations for and against accepting a certain argument. These two cases should be distinguishable, since their outcomes may be different.

3.1 Four-valued paraconsistent labelings

Item 2 above may serve as a motivation for the following definition.

DEFINITION 3.1
Let $\mathcal{AF}=\langle Args, Att\rangle$ be an argumentation framework. A *four-valued labeling* for $\mathcal{AF}$ is a complete function $lab: Args\rightarrow\{\mathsf{in},\mathsf{out},\mathsf{none},\mathsf{both}\}$. We shall sometimes write $\mathsf{None}(lab)$ for $\{A\in Args \mid lab(A)=\mathsf{none}\}$ and $\mathsf{Both}(lab)$ for $\{A\in Args\mid lab(A)=\mathsf{both}\}$.

As before, a labeling function reflects the state of mind of the reasoner regarding each argument in $\mathcal{AF}$. The difference is, of-course, that four-valued labelings are a refinement of 'standard' labelings (in the sense of Definition 2.3), so that *four* states are allowed. Thus, we continue to denote by $\mathsf{In}(lab)$ the set of arguments that one accepts and by $\mathsf{Out}(lab)$ the set of arguments that one rejects, but now the set $\mathsf{Undec}(lab)$ is splitted to two new sets: $\mathsf{None}(lab)$, consisting of arguments that may neither be accepted nor rejected, and $\mathsf{Both}(lab)$, consisting of arguments that have both supportive

[3]Works on the relations between Dung's-style extensions and (partial) status assignments may be traced back to [32].

FIGURE 1. The argumentation framework $\mathcal{AF}_1$.

and rejective evidences. Since a four-valued labeling *lab* is a partition of *Args*, we shall sometimes write it as a quadruple $\langle \mathsf{In}(lab), \mathsf{Out}(lab), \mathsf{None}(lab), \mathsf{Both}(lab) \rangle$.

DEFINITION 3.2
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework.

- Given a set $Ext \subseteq Args$ of arguments, the function that is *induced by* (or, is *associated with*) *Ext* is the four-valued labeling $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ of $\mathcal{AF}$,[4] defined for every $A \in Args$ as follows:

$$p\mathcal{EL}_{\mathcal{AF}}(Ext)(A) = \begin{cases} \mathsf{in} & \text{if } A \in Ext \text{ and } A \notin Ext^+, \\ \mathsf{both} & \text{if } A \in Ext \text{ and } A \in Ext^+, \\ \mathsf{out} & \text{if } A \notin Ext \text{ and } A \in Ext^+, \\ \mathsf{none} & \text{if } A \notin Ext \text{ and } A \notin Ext^+. \end{cases}$$

  A four-valued labeling that is induced by some subset of *Args* is called a *paraconsistent labeling* (or a *p-labeling*) of $\mathcal{AF}$.
- Given a four-valued labeling *lab* of $\mathcal{AF}$, the set of arguments that is *induced by* (or, is *associated with*) *lab* is defined by

$$p\mathcal{LE}_{\mathcal{AF}}(lab) = \mathsf{In}(lab) \cup \mathsf{Both}(lab).$$

The intuition behind the transformation from a labeling *lab* to its extension $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is that any argument for which there is some supportive indication (i.e. it is labeled in or both) should be included in the extension (even if there are also opposing indications). The transformation from an extension *Ext* to its induced labeling function $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is motivated by the aspiration to accept the arguments in the extension by marking them as either in or both. Since *Ext* is not necessarily conflict-free, two labels are required to indicate whether the argument at hand is attacked by another argument in the extension, or not.

Definition 3.2 indicates a one-to-one correspondence between sets of arguments of an abstract argumentation framework and the labelings that are induced by them. It follows that while there are $4^{|Args|}$ four-valued labelings for an argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$, the number of paraconsistent labelings (p-labelings) for $\mathcal{AF}$ is limited by the number of the subsets of *Args*, i.e. $2^{|Args|}$.

EXAMPLE 3.3
Consider the argumentation framework $\mathcal{AF}_1$ of Figure 1. To compute the paraconsistent labelings of $\mathcal{AF}_1$, note for instance that if for some $Ext \subseteq Args$ it holds that $p\mathcal{EL}_{\mathcal{AF}}(Ext)(A) = \mathsf{in}$, then $A \in Ext$ and $A \notin Ext^+$, which implies, respectively, that $B \in Ext^+$ and $B \notin Ext$, thus B must be labeled out. Similarly, if A is labeled out then B must be labeled in, if A is labeled both, B must be labeled both as well, and if A is labeled none, so B is labeled none. These labelings correspond to the four possible choices of either accepting exactly one of the mutually attacking arguments A and B, accepting both of them, or rejecting both of them. In turn, each such choice is augmented with four respective options for labeling C and D. Table 1 lists the corresponding sixteen p-labelings of $\mathcal{AF}_1$.

[4] Here, $p\mathcal{EL}$ stands for a **p**araconsistent-based conversion of **e**xtensions to **l**abelings.

TABLE 1. The p-labelings of $\mathcal{AF}_1$

	A	B	C	D	Induced set
1	in	out	in	out	$\{A,C\}$
2	in	out	in	both	$\{A,C,D\}$
3	in	out	none	in	$\{A,D\}$
4	in	out	none	none	$\{A\}$
5	out	in	out	in	$\{B,D\}$
6	out	in	out	none	$\{B\}$
7	out	in	both	out	$\{B,C\}$
8	out	in	both	both	$\{B,C,D\}$
9	none	none	in	out	$\{C\}$
10	none	none	in	both	$\{C,D\}$
11	none	none	none	in	$\{D\}$
12	none	none	none	none	$\{\}$
13	both	both	out	in	$\{A,B,D\}$
14	both	both	out	none	$\{A,B\}$
15	both	both	both	out	$\{A,B,C\}$
16	both	both	both	both	$\{A,B,C,D\}$

A p-labeling may be regarded as a description of the state of affairs for *any* chosen set of arguments in a framework. For instance, the second p-labeling in Table 1 (Example 3.3) indicates that if $\{A,C,D\}$ is the accepted set of arguments, then B is rejected (labeled out) since it is attacked by an accepted argument, and the status of D is ambiguous (so it is labeled both), since on one hand it is included in the set of accepted arguments, but on the other hand it is attacked by an accepted argument (C). Note, further, that choosing D as an accepted argument in this case is somewhat arguable, since D is not defended by the set $\{A,C,D\}$.

The discussion above implies that the role of a p-labeling is *indicative* rather than *justificatory*; A labeling that is induced by *Ext* describes the role of each argument in the framework according to *Ext*, but it does not *justify* the choice of *Ext* as a plausible extension for the framework. For the latter, we should pose further restrictions on the p-labelings. This is what we do next.

DEFINITION 3.4
Let $\mathcal{AF} = \langle Args, Att\rangle$ be an argumentation framework. A p-labeling *lab* for $\mathcal{AF}$ is called *p-admissible*, if it satisfies the following rules:[5]

pIn If $lab(A) = \mathsf{in}$, then for every $B \in A^-$ it holds that $lab(B) = \mathsf{out}$.

pOut If $lab(A) = \mathsf{out}$, then there exists some $B \in A^-$ such that $lab(B) \in \{\mathsf{in}, \mathsf{both}\}$.

pBoth If $lab(A) = \mathsf{both}$, then for every $B \in A^-$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{both}\}$ and there exists some $B \in A^-$ such that $lab(B) = \mathsf{both}$.

pNone If $lab(A) = \mathsf{none}$, then for every $B \in A^-$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$.

The constraints in Definition 3.4 may be strengthened as follows:

DEFINITION 3.5
Let $\mathcal{AF} = \langle Args, Att\rangle$ be an argumentation framework. A p-labeling *lab* for $\mathcal{AF}$ is called *p-complete*, if it satisfies the following rules:

pIn$^+$ $lab(A) = \mathsf{in}$ iff for every $B \in A^-$ it holds that $lab(B) = \mathsf{out}$.

[5]The notion of p-admissibility should not be confused with a similar notion, used in [20] for prudent semantics.

pOut$^+$ $lab(A) = \mathsf{out}$ iff there is some $B \in A^-$ such that $lab(B) \in \{\mathsf{in}, \mathsf{both}\}$ and there is some $B \in A^-$ such that $lab(B) \in \{\mathsf{in}, \mathsf{none}\}$.

pBoth$^+$ $lab(A) = \mathsf{both}$ iff for every $B \in A^-$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{both}\}$ and there exists some $B \in A^-$ such that $lab(B) = \mathsf{both}$.

pNone$^+$ $lab(A) = \mathsf{none}$ iff for every $B \in A^-$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$ and there exists some $B \in A^-$ such that $lab(B) = \mathsf{none}$.

EXAMPLE 3.6
Consider again the p-labelings for $\mathcal{AF}_1$ (Example 3.3), listed in Table 1.

- The rule **pIn** is violated by labelings 3, 9, 10, 11, and the rule **pBoth** is violated by labelings 2, 7, 8, 10. Therefore, the p-admissible labelings in this case are 1, 4, 5, 6, 12–16.
- Among the p-admissible labelings in the previous item, labelings 4 and 6 violate **pNone**$^+$, and labelings 13–15 violate **pOut**$^+$. Thus, the p-complete labelings of $\mathcal{AF}_1$ are 1, 5, 12 and 16.[6]

In Sections 3.3 and 4, we shall justify the rules in Definitions 3.4 and 3.5 by showing the correspondence between p-admissible/p-complete labelings and related extensions.

NOTE 3.7
Four-valued labeling for abstract argumentation frameworks has already been considered by Jakobovits and Vermeir in [26].[7] Their motivation and goals are different though, which leads to different types of semantics than the present ones. According to [26], using our notations, the four possible labels intuitively indicate acceptance (in), rejection (out), undecided positions (both), and don't-care states (none). The intuitive understanding of both as indicating neither acceptance nor rejection imply, in particular, that accepted arguments are only those with in-labels, and that the underlying semantics of [26] accepts only conflict-free sets.[8]

3.2 *Paraconsistent extensions*

Recall that Item 1 at the beginning of Section 3 suggests that the 'conflict-freeness' requirement in Definition 2.2 may be abandoned. However, the other properties in the same definition, implying that an argument in an extension must be defended, are still necessary.

DEFINITION 3.8
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework and let $Ext \subseteq Args$.

- *Ext* is a *paraconsistently admissible* (or: *p-admissible*) extension for $\mathcal{AF}$, if $Ext \subseteq \mathrm{Def}(Ext)$.
- *Ext* is a *paraconsistently complete* (or: *p-complete*) extension for $\mathcal{AF}$, if $Ext = \mathrm{Def}(Ext)$.

Thus, every admissible (respectively, complete) extension for $\mathcal{AF}$ is also p-admissible (respectively, p-complete) extension for $\mathcal{AF}$, but not the other way around.

EXAMPLE 3.9
The argumentation framework $\mathcal{AF}_2$ that is shown in Figure 2 has two p-complete extensions: $\emptyset$ (which is also the only complete extension of $\mathcal{AF}_2$), and $\{A, B, C\}$.

[6] Intuitively, these labelings represent the most plausible states corresponding to the four possible choices of arguments out of the mutually attacking arguments *A* and *B*.

[7] Martin Caminada is thanked for pointing this out.

[8] We note, however, that in [26] some primary notions (like definability) differ from those of Dung's theory (given in Section 2), and are based on sets of arguments that are not necessarily conflict-free.

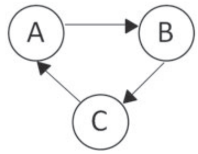

FIGURE 2. The argumentation framework $\mathcal{AF}_2$.

It is well-known that every argumentation framework has at least one complete extension. However, there are cases (e.g. the argumentation framework $\mathcal{AF}_2$ in Figure 2) that the only complete extension for a framework is the empty set. The next proposition shows that this is not the case regarding p-complete extensions.

PROPOSITION 3.10
Any argumentation framework has a non-empty p-complete extension.

PROOF. Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Suppose first that there is an argument $A \in Args$ that is not attacked by any other argument (i.e. $A^- = \emptyset$). In this case, it is well known that the set consisting of all the non-attacked arguments, as well as the arguments that are directly or indirectly defended by non-attacked arguments, is the unique grounded extension of $\mathcal{AF}$. As such, this set is in particular a complete extension of $\mathcal{AF}$, and so it is a p-complete extension of $\mathcal{AF}$.

Suppose now that every argument in *Args* is attacked. We show that in this case *Args* itself is a p-complete extension of $\mathcal{AF}$. Indeed, trivially $\mathrm{Def}(Args) \subseteq Args$, since any set of arguments is a subset of *Args*. For the converse, let $A \in Args$, and let $B \in A^-$. Since $B \in Args$, $B^- \neq \emptyset$, and so $B \in Args^+$. It follows that $A^- \subseteq Args^+$, and so $A \in \mathrm{Def}(Args)$. This shows that $Args \subseteq \mathrm{Def}(Args)$, and so we conclude that $Args = \mathrm{Def}(Args)$. ■

3.3 Relating paraconsistent extensions and paraconsistent labelings

We are now ready to consider the extension-based semantics induced by paraconsistent labelings. We show, in particular, that as in the case of (conflict-free) complete labelings and (conflict-free) complete extensions, there is a one-to-one correspondence between them, thus they represent two equivalent approaches for giving conflict-tolerant semantics to abstract argumentation frameworks.

PROPOSITION 3.11
If *Ext* is a p-admissible extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a p-admissible labeling of $\mathcal{AF}$.

PROOF. Let *Ext* be a p-admissible extension of $\mathcal{AF} = \langle Args, Att \rangle$. Below, we abbreviate $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ by lab_{Ext}.

(1) Suppose that $lab_{Ext}(A) = \mathsf{in}$ and let $B \in A^-$. By the definition of lab_{Ext}, $A \in Ext$. Since *Ext* is p-admissible it defends A, thus $B \in Ext^+$. Also, $A \notin Ext^+$ and so $B \notin Ext$. By the definition of lab_{Ext}, then, $lab_{Ext}(B) = \mathsf{out}$. This shows **pIn**.
(2) Suppose that $lab_{Ext}(A) = \mathsf{out}$. Then $A \in Ext^+$, and so there is $B \in A^-$ such that $B \in Ext$. By the definition of lab_{Ext}, then, $lab_{Ext}(B) \in \{\mathsf{in}, \mathsf{both}\}$. This shows **pOut**.
(3) Suppose that $lab_{Ext}(A) = \mathsf{both}$. As in the case that $lab_{Ext}(A) = \mathsf{in}$, this implies that for every $B \in A^-$, $B \in Ext^+$ as well, and so $lab_{Ext}(B) \in \{\mathsf{out}, \mathsf{both}\}$. Also, $A \in Ext^+$, and so there is a

$B' \in A^-$ such that $B' \in Ext$. For this B', $lab_{Ext}(B') \neq \mathsf{out}$, thus $lab_{Ext}(B') = \mathsf{both}$. This shows **pBoth**.

(4) Suppose that $lab_{Ext}(A) = \mathsf{none}$ and let $B \in A^-$. By the definition of lab_{Ext}, $A \notin Ext^+$, and so $B \notin Ext$, which implies that $lab_{Ext}(B) \in \{\mathsf{out}, \mathsf{none}\}$. This shows **pNone**.

By Items 1–4, then, lab_{Ext} is a p-admissible labeling of $\mathcal{AF}$. ■

PROPOSITION 3.12
If *lab* is a p-admissible labeling of $\mathcal{AF}$ then $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a p-admissible extension for $\mathcal{AF}$.

PROOF. Let $Ext_{lab} = p\mathcal{LE}_{\mathcal{AF}}(lab)$. We have to show that $Ext_{lab} \subseteq \mathrm{Def}(Ext_{lab})$. Indeed, let $A \in Ext_{lab}$. Then $lab(A) \in \{\mathsf{in}, \mathsf{both}\}$. If $A^- = \emptyset$ then obviously $A^- \subseteq Ext^+_{lab}$, and so $A \in \mathrm{Def}(Ext_{lab})$. Suppose then that $A^- \neq \emptyset$.

- If $lab(A) = \mathsf{in}$, then by **pIn**, for every $B \in A^-$ it holds that $lab(B) = \mathsf{out}$. Thus, by **pOut**, for every $B \in A^-$ there is $C \in B^-$ such that $lab(C) \in \{\mathsf{in}, \mathsf{both}\}$, i.e. $C \in Ext_{lab}$. Hence, for every $B \in A^-$ it holds that $B \in Ext^+_{lab}$, and so $A^- \subseteq Ext^+_{lab}$, i.e. $A \in \mathrm{Def}(Ext_{lab})$.
- If $lab(A) = \mathsf{both}$, then by **pBoth**, for every $B \in A^-$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{both}\}$. By **pOut** and **pBoth** this means that for every $B \in A^-$ there is $C \in B^-$ such that $lab(C) \in \{\mathsf{in}, \mathsf{both}\}$. Again, this implies that $B \in Ext^+_{lab}$. We conclude in this case as well that $A^- \subseteq Ext^+_{lab}$, i.e. that $A \in \mathrm{Def}(Ext_{lab})$.

In each case, we have that $A \in \mathrm{Def}(Ext_{lab})$ when $A \in Ext_{lab}$, thus $Ext_{lab} \subseteq \mathrm{Def}(Ext_{lab})$, and so Ext_{lab} is a p-admissible extension for $\mathcal{AF}$. ■

PROPOSITION 3.13
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework.

- For every p-admissible labeling *lab* for $\mathcal{AF}$ it holds that $p\mathcal{EL}_{\mathcal{AF}}(p\mathcal{LE}_{\mathcal{AF}}(lab)) = lab$.
- For every p-admissible extension *Ext* of $\mathcal{AF}$ it holds that $p\mathcal{LE}_{\mathcal{AF}}(p\mathcal{EL}_{\mathcal{AF}}(Ext)) = Ext$.

PROOF. Again, we abbreviate $p\mathcal{LE}_{\mathcal{AF}}(lab)$ by Ext_{lab} and $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ by lab_{Ext}. Now, to show the first item let *lab* be a p-admissible labeling for $\mathcal{AF}$ and let $A \in Args$.

- If $lab(A) = \mathsf{in}$, then by the definition of Ext_{lab} it holds that $A \in Ext_{lab}$. Also, by **pIn**, $A \notin Ext^+_{lab}$ (otherwise, $A \in Ext^+_{lab}$ and so there is $B \in A^-$ such that $lab(B) \in \{\mathsf{in}, \mathsf{both}\}$, which contradicts **pIn**). It follows that $lab_{Ext_{lab}}(A) = \mathsf{in}$, and so $lab_{Ext_{lab}}(A) = lab(A)$.
- If $lab(A) = \mathsf{out}$, then by the definition of Ext_{lab} it holds that $A \notin Ext_{lab}$. Also, by **pOut**, $A \in Ext^+_{lab}$ (otherwise, $A \notin Ext^+_{lab}$ and so for every $B \in A^-$ we have that $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$, which contradicts **pOut**). It follows that $lab_{Ext_{lab}}(A) = \mathsf{out}$, and so $lab_{Ext_{lab}}(A) = lab(A)$.
- If $lab(A) = \mathsf{both}$, then by the definition of Ext_{lab} it holds that $A \in Ext_{lab}$. Also, by **pBoth**, $A \in Ext^+_{lab}$ (otherwise, $A \notin Ext^+_{lab}$ and so for every $B \in A^-$ we have that $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$, which contradicts **pBoth**). It follows that $lab_{Ext_{lab}}(A) = \mathsf{both}$, and so $lab_{Ext_{lab}}(A) = lab(A)$.
- If $lab(A) = \mathsf{none}$, then by the definition of Ext_{lab} it holds that $A \notin Ext_{lab}$. Also, by **pNone**, $A \notin Ext^+_{lab}$ (otherwise, $A \in Ext^+_{lab}$ and so there is $B \in A^-$ such that $lab(B) \in \{\mathsf{in}, \mathsf{both}\}$, which contradicts **pNone**). It follows that $lab_{Ext_{lab}}(A) = \mathsf{none}$, and so $lab_{Ext_{lab}}(A) = lab(A)$.

In each case, then, the labeling of $lab_{Ext_{lab}}$ coincides with that of *lab*, which shows the first item. For the second item, let *Ext* be an extension of $\mathcal{AF}$.

- To see that $Ext \subseteq Ext_{lab_{Ext}}$, let $A \in Ext$. If $A \notin Ext^+$ then $lab_{Ext} = \mathsf{in}$ and so $A \in Ext_{lab_{Ext}}$. Otherwise, $A \in Ext^+$ thus $lab_{Ext} = \mathsf{both}$, and again $A \in Ext_{lab_{Ext}}$.

- To see that $Ext_{lab_{Ext}} \subseteq Ext$, let $A \in Ext_{lab_{Ext}}$. By the definition of $Ext_{lab_{Ext}}$ it holds that either $lab_{Ext} = \mathsf{in}$ or $lab_{Ext} = \mathsf{both}$. In either cases, $A \in Ext$.

It follows that $Ext_{lab_{Ext}} = Ext$, as required. ■

NOTE 3.14
The requirement in the second item of the last proposition, that *Ext* should be p-admissible, is for the analogy with the first item of the same proposition (and for Corollary 3.15 below), but it is not really necessary for the proof.

By Propositions 3.11, 3.12 and 3.13, we conclude the following.

COROLLARY 3.15
The functions $p\mathcal{EL}_{\mathcal{AF}}$ and $p\mathcal{LE}_{\mathcal{AF}}$, restricted to the p-admissible labelings and the p-admissible extensions of $\mathcal{AF}$, are bijective, and are each other's inverse.

It follows that p-admissible extensions and p-admissible labelings are, in essence, different ways of describing the same thing (see also Figure 3 below).

NOTE 3.16
In a way, the correspondence between p-admissible extensions and p-admissible labelings of an argumentation framework is tighter than the correspondence between (conflict-free) admissible extensions and (conflict-free) admissible labelings, as depicted in [19] (see Section 2). Indeed, as indicated in [19], admissible labelings and admissible extensions have a many-to-one relationship: each admissible labeling is associated with exactly one admissible extension, but an admissible extension may be associated with several admissible labelings. For instance, in the argumentation framework $\mathcal{AF}_1$ of Figure 1 (Example 3.3), $lab_1 = \langle\{B\},\{A,C\},\{D\}\rangle$ and $lab_2 = \langle\{B\},\{A\},\{C,D\}\rangle$ are different admissible labelings that are associated with the same admissible extension $\{B\}$. Note that, viewed as *four-valued* labelings into $\{\mathsf{in},\mathsf{out},\mathsf{none}\}$, only lab_1 is p-admissible, since lab_2 violates **pNone**. Indeed, the p-admissible extension $\{B\}$ is associated with exactly *one* p-admissible labeling (number 6 in Table 1), as guaranteed by the last corollary.

Let us now consider p-complete labelings.

PROPOSITION 3.17
If *Ext* is a p-complete extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a p-complete labeling of $\mathcal{AF}$.

PROOF. Let *Ext* be a p-complete extension of $\mathcal{AF}$. Again, we abbreviate $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ by lab_{Ext}. Note, first, that since *Ext* is in particular a p-admissible extension of $\mathcal{AF}$, by Proposition 3.11 lab_{Ext} is a p-admissible labeling of $\mathcal{AF}$, so it satisfies **pIn**, **pOut**, **pBoth** and **pNone**.

(1) One direction of **pIn**$^+$ is the rule **pIn**, shown in the proof of Proposition 3.12 for p-admissible extensions, so it certainly holds for p-complete extensions. For the converse, suppose that $lab_{Ext}(B) = \mathsf{out}$ for every $B \in A^-$. By the definition of lab_{Ext}, then, $B \in Ext^+$ for every $B \in A^-$, and so $A^- \subseteq Ext^+$. Thus, $A \in \mathrm{Def}(Ext)$. But *Ext* is p-complete, hence $A \in Ext$. It follows that $lab_{Ext}(A) \in \{\mathsf{in},\mathsf{both}\}$. By **pBoth**, $lab_{Ext}(A) \neq \mathsf{both}$, since there is no $B \in A^-$ such that $lab_{Ext}(B) = \mathsf{both}$. Thus $lab_{Ext}(A) = \mathsf{in}$. This shows **pIn**$^+$.
(2) Suppose that $lab_{Ext}(A) = \mathsf{out}$. By **pOut** (which holds already for p-admissible extensions), there is some $B \in A^-$ such that $lab_{Ext}(B) \in \{\mathsf{in},\mathsf{both}\}$. Furthermore, it is not possible that for *every* $B \in A^-$ it holds that $lab_{Ext}(B) \in \{\mathsf{both},\mathsf{out}\}$, since in that case $A^- \subseteq Ext^+$, which implies that $A \in \mathrm{Def}(Ext)$ and since *Ext* is p-complete, $A \in Ext$. This contradicts the fact that $lab_{Ext}(A) = \mathsf{out}$. Therefore there is some $B \in A^-$ such that $lab_{Ext}(B) \in \{\mathsf{in},\mathsf{none}\}$.

For the converse, note that if there is some $B\in A^-$ such that $lab_{Ext}(B)\in\{\mathsf{in},\mathsf{none}\}$, then by **pIn**, $lab_{Ext}(A)\neq\mathsf{in}$, and by **pBoth**, $lab_{Ext}(A)\neq\mathsf{both}$. Moreover, since there is some $B\in A^-$ such that $lab_{Ext}(B)\in\{\mathsf{in},\mathsf{both}\}$, by **pNone**, $lab_{Ext}(A)\neq\mathsf{none}$. Thus, necessarily $lab_{Ext}(A)=\mathsf{out}$. This shows **pOut**$^+$.

(3) Suppose that $lab_{Ext}(A)=\mathsf{none}$. By **pNone** (which holds already for p-admissible extensions), for every $B\in A^-$ it holds that $lab_{Ext}(B)\in\{\mathsf{out},\mathsf{none}\}$. By **pIn**$^+$ it is not possible that $lab_{Ext}(B)=\mathsf{out}$ for every $B\in A^-$ (otherwise $lab_{Ext}(A)=\mathsf{in}$), and so there is some $B\in A^-$ such that $lab_{Ext}(B)=\mathsf{none}$.
For the converse, suppose that there exists some $B\in A^-$ such that $lab_{Ext}(B)=\mathsf{none}$. By **pIn**$^+$, then, $lab_{Ext}(A)\neq\mathsf{in}$ and by **pBoth**, $lab_{Ext}(A)\neq\mathsf{both}$. Also, since for every $B\in A^-$ it holds that $lab_{\mathcal{AF}}(Ext)(B)\in\{\mathsf{out},\mathsf{none}\}$, by **pOut**, $lab_{Ext}(A)\neq\mathsf{out}$. Therefore, necessarily $lab_{Ext}(A)=\mathsf{none}$. This shows **pNone**$^+$.

(4) One direction of **pBoth**$^+$ is the rule **pBoth**, shown in the proof of Proposition 3.12 for p-admissible extensions, so it certainly holds for p-complete extensions. For the converse, suppose that for every $B\in A^-$ it holds that $lab_{Ext}(B)\in\{\mathsf{out},\mathsf{both}\}$ and there exists some $B\in A^-$ such that $lab_{Ext}(B)=\mathsf{both}$. The first condition together with **pNone**$^+$ show that $lab_{Ext}(A)\neq\mathsf{none}$ (since there is no $B\in A^-$ such that $lab_{Ext}(B)=\mathsf{none}$), and together with **pOut**$^+$, $lab_{Ext}(A)\neq\mathsf{out}$ (since there is no $B\in A^-$ such that $lab_{Ext}(B)\in\{\mathsf{in},\mathsf{none}\}$). The second condition, together with **pIn**$^+$ show that $lab_{Ext}(A)\neq\mathsf{in}$ (since it is not the case that for all $B\in A^-$, $lab_{Ext}(B)=\mathsf{out}$). Thus, necessarily $lab_{Ext}(A)=\mathsf{both}$. This shows **pBoth**$^+$.

By Items 1–4, then, lab_{Ext} is a p-complete labeling of $\mathcal{AF}$. ■

Proposition 3.18
If lab is a p-complete labeling of $\mathcal{AF}$ then $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a p-complete extension for $\mathcal{AF}$.

Proof. Let $Ext_{lab}=p\mathcal{LE}_{\mathcal{AF}}(lab)$. We have to show that $Ext_{lab}=\mathrm{Def}(Ext_{lab})$. By Proposition 3.12, since lab is in particular p-admissible, $Ext_{lab}\subseteq\mathrm{Def}(Ext_{lab})$. It remains to show, then, that $\mathrm{Def}(Ext_{lab})\subseteq Ext_{lab}$. Indeed, let $A\in\mathrm{Def}(Ext_{lab})$. We show that this implies that $lab(A)\in\{\mathsf{in},\mathsf{both}\}$, and so $A\in Ext_{lab}$. First, note that if $A^-=\emptyset$, then by **pIn**$^+$, $lab(A)=\mathsf{in}$. So in what follows we assume that $A^-\neq\emptyset$. Now,

- If $lab(A)=\mathsf{out}$ then by **pOut**$^+$ there is $B\in A^-$ such that $lab(B)\in\{\mathsf{in},\mathsf{none}\}$, and since lab is a p-labeling this means that $B\notin Ext^+_{lab}$. It follows that $A^-\not\subseteq Ext^+_{lab}$ and so $A\notin\mathrm{Def}(Ext_{lab})$, a contradiction to our assumption.
- If $lab(A)=\mathsf{none}$ then by **pNone**$^+$ there is $B\in A^-$ so that $lab(B)=\mathsf{none}$, and since lab is a p-labeling this means that $B\notin Ext^+_{lab}$. Again, this implies that $A^-\not\subseteq Ext^+_{lab}$ and so $A\notin\mathrm{Def}(Ext_{lab})$, which contradicts our assumption about A.

In each case above we reached a contradiction, thus $lab(A)\in\{\mathsf{in},\mathsf{both}\}$, and so $A\in Ext_{lab}$. In conclusion, then, $Ext_{lab}=\mathrm{Def}(Ext_{lab})$, and so Ext_{lab} is a p-complete extension for $\mathcal{AF}$. ■

Proposition 3.19
Let $\mathcal{AF}$ be an argumentation framework.

- For every p-complete labeling lab for $\mathcal{AF}$ it holds that $p\mathcal{EL}_{\mathcal{AF}}(p\mathcal{LE}_{\mathcal{AF}}(lab))=lab$.
- For every p-complete extension Ext of $\mathcal{AF}$ it holds that $p\mathcal{LE}_{\mathcal{AF}}(p\mathcal{EL}_{\mathcal{AF}}(Ext))=Ext$.

Proof. By Proposition 3.13 and the fact that every p-complete labeling for $\mathcal{AF}$ (respectively, p-complete extension of $\mathcal{AF}$) is also a p-admissible labeling for $\mathcal{AF}$ (respectively, p-admissible extension of $\mathcal{AF}$). ■

By Propositions 3.17, 3.18 and 3.19, we conclude the following.

Corollary 3.20
The functions $p\mathcal{EL}_{\mathcal{AF}}$ and $p\mathcal{LE}_{\mathcal{AF}}$, restricted to the p-complete labelings and the p-complete extensions of $\mathcal{AF}$, are bijective, and are each other's inverse.

It follows that p-complete extensions and p-complete labelings are different ways of describing the same thing (see also Figure 3). This is in correlation with the results for conflict-free semantics, according to which there is a one-to-one relationship between complete extensions and complete labelings (Proposition 2.5).

Example 3.21
Consider again the framework $\mathcal{AF}_1$ of Example 3.3.

(1) By Example 3.6 and Propositions 3.11, 3.12, the p-admissible extensions of $\mathcal{AF}_1$ are those that are induced by labelings 1, 4, 5, 6, 12–16 in Table 1, i.e. $\{A,C\}$, $\{A\}$, $\{B,D\}$, $\{B\}$, $\emptyset$, $\{A,B,D\}$, $\{A,B\}$, $\{A,B,C\}$, and $\{A,B,C,D\}$ (respectively).
(2) By Example 3.6 and Propositions 3.17, 3.18, the p-complete extensions of $\mathcal{AF}_1$ are those that are induced by labelings 1, 5, 12 and 16 in Table 1, namely $\{A,C\}$, $\{B,D\}$, $\emptyset$, and $\{A,B,C,D\}$ (respectively).

4 From conflict-tolerant to conflict-free semantics

In this section we show that the variety of 'standard' semantics for argumentation frameworks, based on conflict-free extensions and conflict-free labeling functions, are still available in our conflict-tolerant setting. First, we consider admissible extensions (Definition 2.2) and admissible labelings (Definition 2.4).

Proposition 4.1
Let *lab* be a p-admissible labeling for an argumentation framework $\mathcal{AF}$. If *lab* is into {in, out, none},[9] then it is admissible.

Proof. Let *lab* be a p-admissible labeling. Then in particular it satisfies **pIn**, and so it satisfies **Pos2**.[10] Furthermore, since *lab* satisfies **pOut** and it is both-free, it necessarily satisfies **Neg**, and so *lab* is admissible in the sense of Definition 2.4. ∎

As Note 3.16 shows, the converse of the last proposition does not hold. Indeed, as indicated by Caminada and Gabbay [19], there is a many-to-one relationship between admissible labelings and admissible extensions. On the other hand, by the next proposition (together with Corollary 3.15), there is a one-to-one relationship between both-free p-admissible labelings and admissible extensions.

Proposition 4.2
Let $\mathcal{AF} = \langle Args, Att\rangle$ be an argumentation framework. Then

(1) If *lab* is a both-free p-admissible labeling for $\mathcal{AF}$, then $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is an admissible extension of $\mathcal{AF}$.
(2) If *Ext* is an admissible extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a both-free p-admissible labeling for $\mathcal{AF}$.

[9] In which case *lab* is called 'both-free'.

[10] We use different notations for these rules to emphasize that the former applies to four-valued labelings while the latter applies to three-valued labelings.

PROOF. Item 1 follows from the fact that since *lab* is a p-admissible labeling, then by Proposition 3.12 $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a p-admissible extension of $\mathcal{AF}$, thus $p\mathcal{LE}_{\mathcal{AF}}(lab) \subseteq \mathrm{Def}(p\mathcal{LE}_{\mathcal{AF}}(lab))$. Moreover, since *lab* is both-free, there is no $A \in p\mathcal{LE}_{\mathcal{AF}}(lab) \cap p\mathcal{LE}_{\mathcal{AF}}(lab)^{+}$, thus $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is conflict-free. It follows, then, that $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is an admissible extension of $\mathcal{AF}$.

Item 2 follows from the fact that since *Ext* is an admissible extension of $\mathcal{AF}$, it is in particular p-admissible extension of $\mathcal{AF}$, and so by Proposition 3.11, $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a p-admissible labeling for $\mathcal{AF}$. Moreover, since *Ext* is conflict-free, there is no argument $A \in Args$ such that both $A \in Ext$ and $A \in Ext^{+}$. This implies that $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is both-free. ∎

NOTE 4.3
In [19] Caminada and Gabbay consider also JV-labeling (which actually goes back to [26]), and show that such labelings are in a one-to-one correspondence with admissible extensions. Thus, by the last proposition, JV-labelings are in a one-to-one correspondence with both-free p-admissible labelings.[11]

Let us now consider complete extensions and complete labelings. The next two propositions are the analogue, for complete labelings and complete extensions, of Propositions 4.1 and 4.2.

PROPOSITION 4.4
Let *lab* be a p-complete labeling for an argumentation framework $\mathcal{AF}$. If *lab* is into $\{\mathsf{in}, \mathsf{out}, \mathsf{none}\}$, then it is complete.

PROOF. Suppose that *lab* is p-complete and both-free. Then it is in particular p-admissible, and so by Proposition 4.1 it is admissible. To show that *lab* is complete in the sense of Definition 2.4 it therefore remains to show that *lab* satisfies **Neither** (when the label undec is renamed by none). Indeed, suppose that $lab(A) = \mathsf{none}$. By $\mathbf{pNone}^{+}$, for every $B \in A^{-}$ it holds that $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$ and there exists some $B \in A^{-}$ such that $lab(B) = \mathsf{none}$. Thus, not for every $B \in A^{-}$ it holds that $lab(B) = \mathsf{out}$ and there does not exist a $B \in A^{-}$ such that $lab(B) = \mathsf{in}$. ∎

PROPOSITION 4.5
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Then

(1) If *lab* is a both-free p-complete labeling for $\mathcal{AF}$, then $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete extension of $\mathcal{AF}$.
(2) If *Ext* is a complete extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a both-free p-complete labeling for $\mathcal{AF}$.

PROOF. Suppose that *lab* is a both-free p-complete labeling for $\mathcal{AF}$. By Proposition 4.4, *lab* is a complete labeling of $\mathcal{AF}$. Thus, by Proposition 2.5, $\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete extension of $\mathcal{AF}$. But since *lab* is both-free, we have that $\mathcal{LE}_{\mathcal{AF}}(lab) = p\mathcal{LE}_{\mathcal{AF}}(lab)$, thus $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete extension of $\mathcal{AF}$.

Suppose now that *Ext* is a complete extension of $\mathcal{AF}$. Then in particular *Ext* is conflict-free, and so $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ cannot have both-assignments (if $p\mathcal{EL}_{\mathcal{AF}}(Ext)(A) = \mathsf{both}$ for some $A \in Args$ then $A \in Ext \cap Ext^{+}$ and so $Ext \cap Ext^{+} \neq \emptyset$). Moreover, since *Ext* is a complete extension of $\mathcal{AF}$, $\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete labeling of $\mathcal{AF}$ (Proposition 2.5 again), and since $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is into $\{\mathsf{in}, \mathsf{out}, \mathsf{none}\}$, it coincides with $\mathcal{EL}_{\mathcal{AF}}(Ext)$. It follows that $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ satisfies the conditions **Pos2**, **Neg**, and **Neither** (when the label undec is replaced by none) in Definition 2.4. This, together with the fact that $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is both-free, implies that $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ satisfies all the conditions in

[11] An anonymous reviewer is acknowledged for indicating this.

Definition 3.5. Indeed, **pBoth**$^+$ is not relevant since the labeling is both-free, **pIn**$^+$ is the same as **Pos2**, and since there are no both-assignments, **pOut**$^+$ and **pNone**$^+$ are the same, respectively, as **Neg**, and **Neither**. Thus, $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a p-complete labeling for $\mathcal{AF}$. ■

PROPOSITION 4.6
Let $\mathcal{AF}$ be an argumentation framework. Then *lab* is a complete labeling for $\mathcal{AF}$ iff it is a both-free p-complete labeling for $\mathcal{AF}$.

PROOF. One direction is shown in Proposition 4.4. For the converse, suppose that *lab* is a complete labeling for $\mathcal{AF}$. Then by Proposition 2.5, $\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete extension of $\mathcal{AF}$, and since *lab* is conflict-free, $p\mathcal{LE}_{\mathcal{AF}}(lab) = \mathcal{LE}_{\mathcal{AF}}(lab) = \mathsf{In}(lab)$. It follows, then, that $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a complete extension of $\mathcal{AF}$. Now, by Item 2 of Proposition 4.5, $p\mathcal{EL}_{\mathcal{AF}}(p\mathcal{LE}_{\mathcal{AF}}(lab))$ is a both-free p-complete labeling for $\mathcal{AF}$. By the first item of Proposition 3.19, then, *lab* is a both-free p-complete labeling for $\mathcal{AF}$. ■

Figure 3 summarizes the relations between the conflict-free semantics and the conflict-tolerant semantics considered so far, as well as the relations between the corresponding extension-based and labeling-based semantics. The arrows in the figure denote 'is-a' relationships, and the double-arrows denote one-to-one relationships. For clarity, some arrows are omitted from the figure. For instance, complete extensions are p-complete extensions, admissible extensions are p-admissible extensions, and similar relations hold for their dual labelings.

By Proposition 4.6, a variety of conflict-free, extension-based (Dung-style) semantics for abstract argumentation frameworks may be defined in terms of both-free p-complete labelings. For instance,

- *Ext* is a grounded extension of $\mathcal{AF}$ iff it is induced by a both-free p-complete labeling *lab* of $\mathcal{AF}$ such that there is no both-free p-complete labeling *lab*$'$ of $\mathcal{AF}$ with $\mathsf{In}(lab') \subset \mathsf{In}(lab)$.

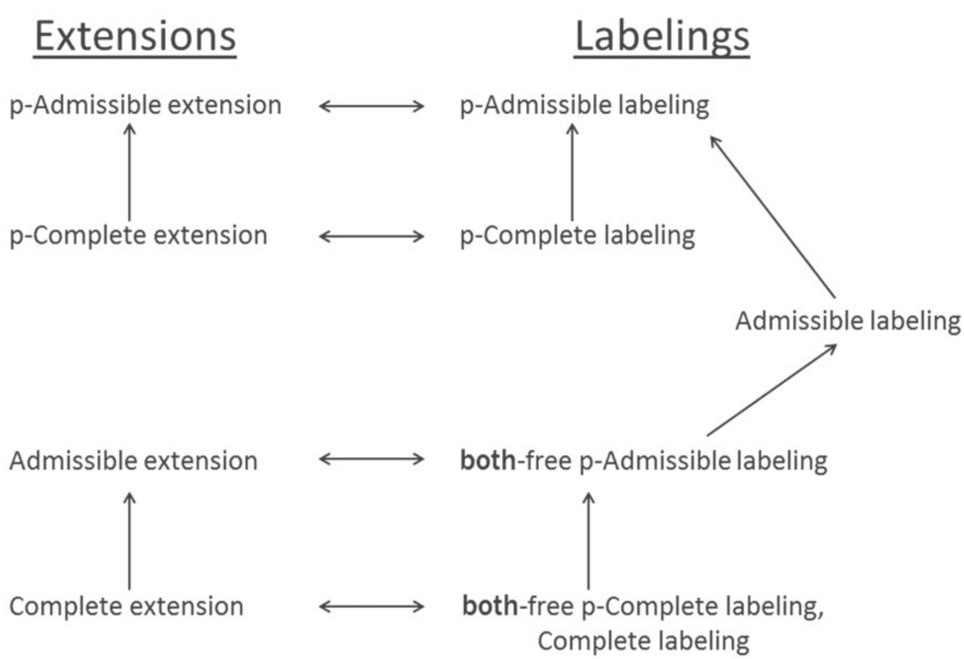

FIGURE 3. Conflict-free and conflict-tolerant semantics.

- *Ext* is a preferred extension of $\mathcal{AF}$ iff it is induced by a both-free p-complete labeling *lab* of $\mathcal{AF}$ such that there is no both-free p-complete labeling lab' of $\mathcal{AF}$ with $\mathsf{In}(lab) \subset \mathsf{In}(lab')$.
- *Ext* is a semi-stable extension of $\mathcal{AF}$ iff it is induced by a both-free p-complete labeling *lab* of $\mathcal{AF}$ such that there is no both-free p-complete labeling lab' of $\mathcal{AF}$ with $\mathsf{None}(lab') \subset \mathsf{None}(lab)$.
- *Ext* is a stable extension of $\mathcal{AF}$ iff it is induced by a both-free p-complete labeling *lab* of $\mathcal{AF}$ such that $\mathsf{None}(lab) = \emptyset$.

By the last item, stable extensions correspond to $\{\mathsf{both}, \mathsf{none}\}$-free p-complete labelings:

Corollary 4.7
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework.

(1) If *lab* is a $\{\mathsf{both}, \mathsf{none}\}$-free p-complete labeling for $\mathcal{AF}$, then $p\mathcal{LE}_{\mathcal{AF}}(lab)$ is a stable extension of $\mathcal{AF}$.
(2) If *Ext* is a stable extension of $\mathcal{AF}$, then $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a $\{\mathsf{both}, \mathsf{none}\}$-free p-complete labeling for $\mathcal{AF}$.

Proof. By Proposition 4.6 and the fact that three- (respectively, four-)valued labelings that are induced by stable extensions are (p-)complete and do not have undec (respectively, none) assignments. ■

Example 4.8
Consider again the framework $\mathcal{AF}_1$ of Example 3.3.

(1) By Proposition 4.5, the complete extensions of $\mathcal{AF}_1$ are those induced by the both-free p-complete labelings, i.e. $\{A, C\}$, $\{B, D\}$ and $\emptyset$ (which are the both-free labelings among those mentioned in Items 2 of Examples 3.6 and 3.21).
(2) By Corollary 4.7, the stable extensions of $\mathcal{AF}_1$ are those induced by the none-free labelings among the labeling in the previous item, i.e. $\{A, C\}$ and $\{B, D\}$.

5 Application: constrained argumentation frameworks

As observed, e.g. by Amgoud and Cayrol [2], Coste-Marquis *et al*. [21], Modgil [27], and others, it is sometimes useful to express some meta-knowledge about the arguments at hand (using, e.g. preferences relations on the arguments) for having a better understanding of the domain of the framework. However, it may happen that such an additional information increases the level of inconsistency of the whole system. In this section, we demonstrate how conflict-tolerant approaches to argumentation semantics may help to handle such situations.

Suppose, for instance, that each argument is equipped with a quantitative measurement, reflecting its plausibility. In the extreme case, such a measurement may indicate that the argument to which it is attached must be accepted, in which case that argument serves as a kind of integrity constraint which must always be taken into account. Thus, the set of constraints consists of arguments that must be included in every extension of the argumentation framework.

A natural requirement from a set of constraints is that it should be p-admissible. This is so, since any accepted argument, not to mention ones that *must* be accepted, has to be justified, and so such arguments should not be exposed to undefended attacks. On the other hand, requiring conflict-freeness from a set of constraints may be too strong.[12] Clearly, if the set of constraints is not conflict-free (i.e. if there are mutual attacks among arguments that must be accepted), no conflict-free extension satisfies the constraints, and so conflict-tolerant semantics is called for.

Definition 5.1
A *constrained argumentation framework* is a triple $\mathcal{CAF} = \langle Args, Att, Const \rangle$, where $\langle Args, Att \rangle$ is an argumentation framework, and *Const* (the set of *constraints*) is a p-admissible subset of *Args*.[13]

Definition 5.2
An admissible (respectively, complete, p-admissible, p-complete) extension for a constrained argumentation framework $\mathcal{CAF} = \langle Args, Att, Const \rangle$ is a superset of *Const*, which is an admissible (respectively, complete, p-admissible, p-complete) extension of $\langle Args, Att \rangle$.

Example 5.3
Consider the constrained argumentation framework $\mathcal{CAF}_1 = \langle Args, Att, Const \rangle$, where $\mathcal{AF}_1 = \langle Args, Att \rangle$ is the argumentation framework of Figure 1 and $Const = \{A, B\}$. This constrained framework does not have admissible nor complete extensions (since *Const* is not conflict-free), but it has four p-admissible extensions: $\{A, B\}$, $\{A, B, C\}$, $\{A, B, D\}$ and $\{A, B, C, D\}$, the latter is also p-complete.

Proposition 5.4
Every constrained argumentation framework has a non-empty p-admissible extension and a nonempty p-complete extension.

Proof. Let $\mathcal{CAF} = \langle Args, Att, Const \rangle$ be a constrained argumentation framework. If $Const = \emptyset$ then $\mathcal{CAF}$ is in fact an ('ordinary') argumentation framework, and so the proposition follows from Proposition 3.10. Suppose then that $Const \neq \emptyset$. By its definition, *Const* is a p-admissible extension of $\mathcal{CAF}$. Now, if *Const* is also a p-complete extension of $\mathcal{CAF}$, we are done. Otherwise, there is an argument $A_1 \in \mathrm{Def}(Const) - Const$, so let $Const_1 = Const \cup \{A_1\}$. Note that $Const_1$ is still p-admissible, since $A_1 \in \mathrm{Def}(Const)$, and so $Const_1 = Const \cup \{A_1\} \subseteq \mathrm{Def}(Const) \subseteq \mathrm{Def}(Const_1)$. Now, if $Const_1$ is p-complete, we are done. Otherwise, we again choose an argument $A_2 \in \mathrm{Def}(Const_1) - Const_1$, and consider the set $Const_2 = Const_1 \cup \{A_2\}$. As before, $Const_2$ is still p-admissible. By this process, we get a sequence of p-admissible extensions $Const, Const_1, Const_2, \ldots$, each extension properly contains the previous one. Note that this sequence consists of no more than $|Args|$ p-admissible sets, and it must culminate in a p-complete extension of $\mathcal{CAF}$. This is so, since if we keep adding arguments without reaching a p-complete extension, we eventually end-up with the whole set of arguments, *Args*. Hence, since the sequence contains only p-admissible extensions, in particular $Args \subseteq \mathrm{Def}(Args)$, and obviously $\mathrm{Def}(Args) \subseteq Args$, thus $Args = \mathrm{Def}(Args)$, and so *Args* is a p-complete extension of $\mathcal{CAF}$.[14] ■

Let *Ext* be a p-admissible or p-complete extension of a constrained argumentation framework $\mathcal{CAF} = \langle Args, Att, Const \rangle$. If *Const* is not conflict-free, the labeling $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ that is induced by *Ext* according to $\mathcal{AF} = \langle Args, Att \rangle$ must have both-assignments. In such cases, a sensible criterion

[12] Recall Example 1.1, for instance, in which the two conflicting arguments about the nature of light may be introduced as constraints.

[13] Alternatively, we shall sometimes refer to a constrained argumentation framework as a *pair* $\langle \mathcal{AF}, Const \rangle$, where $\mathcal{AF}$ is an argumentation framework and *Const* is a p-admissible extension of $\mathcal{AF}$.

[14] Note that the p-complete extension of $\mathcal{CAF}$ constructed in the proof is minimal in the sense that every set that is properly contained in it is not p-complete or does not contain the set *Const*. In this respect, we have shown that $\mathcal{CAF}$ has what may be called a 'p-grounded extension'.

for setting preferences among the p-extensions of $\mathcal{CAF}$ is to choose those whose induced four-valued labelling has a minimal amount of both-assignments. Moreover, when there is ambiguity about arguments in the framework uncertainty can spread throughout the framework, eliminating the possibility to decide on the validity of other arguments. It is therefore desirable to restrict this phenomenon as much as possible. Virtually, then, traditional *conflict-free* extensions of argumentation frameworks are replaced here by *conflict-minimizing* extensions of constrained argumentation frameworks. This is the intuition behind the next definition.

Definition 5.5
Let $\mathcal{CAF}$ be a constrained argumentation framework for an argumentation framework $\mathcal{AF}$ and a set of constraints *Const*. A p-admissible (respectively, p-complete) extension *Ext* of $\mathcal{CAF}$ is *minimally conflicting*, if there is no p-admissible (respectively, p-complete) extension Ext' of $\mathcal{CAF}$ such that

$$\mathsf{Both}(p\mathcal{EL}_{\mathcal{AF}}(Ext')) \subsetneq \mathsf{Both}(p\mathcal{EL}_{\mathcal{AF}}(Ext)).$$ [15]

Example 5.6
Among the four p-admissible extensions of the constrained argumentation framework $\mathcal{CAF}_1$ of Example 5.3, two are minimally conflicting: $\{A,B\}$ and $\{A,B,D\}$.

A criterion for setting further preferences among the minimally conflicting p-admissible (or p-complete) extensions of a constrained argumentation framework could be minimization of the none-assignments. Again, the intuition here is that while uncertainty about certain arguments is sometimes unavoidable, this is usually not desirable and so neutral states should be avoided as much as possible.

Definition 5.7
Let $\mathcal{CAF}$ be a constrained argumentation framework for an argumentation framework $\mathcal{AF}$ and a set of constraints *Const*. A minimally conflicting p-admissible (respectively, p-complete) extension *Ext* of $\mathcal{CAF}$ is *p-semi-stable*, if there is no minimally conflicting p-admissible (respectively, p-complete) extension Ext' of $\mathcal{CAF}$ such that

$$\mathsf{None}(p\mathcal{EL}_{\mathcal{AF}}(Ext')) \subsetneq \mathsf{None}(p\mathcal{EL}_{\mathcal{AF}}(Ext)).$$

Example 5.8
Among the two minimally conflicting p-admissible extensions of the constrained argumentation framework $\mathcal{CAF}_1$ of Example 5.6, only $\{A,B,D\}$ is p-semi-stable. This may be intuitively understood as keeping track to the two conflicting arguments (A and B), as dictated by the constraints, and accepting the argument (D) that is not directly related to the conflict.

Methods for representing and computing minimally conflicting p-extensions of constrained argumentation frameworks will be described in the next section.

Note 5.9
Obviously, minimally conflicting p-complete extensions and p-semi-stable extensions dismiss any conflict that is not introduced by the constraints. Moreover, when the conflicts can be 'isolated' from the rest of the framework, they are 'localized' by these extensions. This happens, e.g. when the argumentation framework at hand can be partitioned into two distinct (non-connected) subgraphs $\mathcal{AF}' = \langle Args', Att' \rangle$ and $\mathcal{AF}'' = \langle Args'', Att'' \rangle$,[16] so that $Args'' \cap Const = \emptyset$. In such cases, if *Ext* is a minimally conflicting p-extension, then $Ext \cap Args''$ is conflict-free. A simple example of this is

[15]Equivalently, $Ext' \cap Ext'^{+} \subsetneq Ext \cap Ext^{+}$.

[16]That is, $(Args')^{+} \cap Args'' = \emptyset$ and $(Args'')^{+} \cap Args' = \emptyset$.

FIGURE 4. The argumentation framework $\mathcal{AF}_3$.

the constrained argumentation framework $\mathcal{CAF}_3 = \langle Args, Att, Const \rangle$, where $\mathcal{AF}_3 = \langle Args, Att \rangle$ is the argumentation framework of Figure 4 and $Const = \{A, B\}$. Here, the minimally conflicting p-complete extensions are $\{A, B\}$, $\{A, B, C\}$ and $\{A, B, D\}$, where the two latter are also p-semi-stable. In neither of them both C and D are accepted.

NOTE 5.10
The introduction of constraints in abstract argumentation frameworks may be useful, e.g., for enforcing reflexivity of entailment relations in the context of Besnard and Hunter's approach to deductive argumentation [13, 14].[17] According to this approach, given a finite set Δ of propositional formulas (the underlying knowledge-base), an argument is a pair $\langle S, \psi \rangle$, where S (the support set) is a classically consistent subset of Δ that is minimal with respect to set inclusion and classically entails the formula ψ (the conclusion). Denote by $Args(\Delta)$ the set of arguments that are constructed from Δ as described above. A corresponding attack relation Att on $Args(\Delta)$ is usually required to meet the following conditions (see [1]):

- *Conflict sensitivity*: If the union of the support sets of two arguments is inconsistent, then at least one of these arguments attacks the other.
- *Conflict dependence*: If an argument attacks another argument, then the union of their support sets is inconsistent.

Intuitively, the above two principles assure, respectively, that all the inconsistencies in Δ are captured by Att and that no attacks belong to Att unless they are reflected in Δ.

Now, a Dung-style argumentation framework (induced by Δ) is a pair $\mathcal{AF}(\Delta) = \langle Args(\Delta), Att \rangle$. Accordingly, extensions of $\mathcal{AF}(\Delta)$ may be used for defining the conclusions of Δ: ψ follows from Δ according to an argumentation semantics S (notation: $\Delta \mid\!\sim_{\mathsf{S}} \psi$), if ψ is the conclusion of an argument that belongs to every S-extension of $\mathcal{AF}(\Delta)$.

Note that by conflict sensitivity the entailment relation $\mid\!\sim_{\mathsf{S}}$ cannot be reflexive when S is based on conflict-free extensions. Indeed, consider for instance the set $\Delta = \{p, \neg p\}$. In this case, conflict sensitivity dictates that at least one of the arguments $A_1 = \langle \{p\}, p \rangle$ or $A_2 = \langle \{\neg p\}, \neg p \rangle$ attacks the other, and so no conflict-free extension of $\mathcal{AF}(\Delta)$ contains both of these arguments. This means, in particular, that at least one of p or $\neg p$ cannot be a $\mid\!\sim_{\mathsf{S}}$-consequence of Δ.

The ability to conclude every premise is a primary principle in many logic-based systems (in particular those that are based on Tarskian consequence relations [31], where reflexivity is an explicit requirement). In our case, this property can be sometimes guaranteed (on the expense of keeping the set of conclusions classically consistent) by including Δ in the set of constraints. Indeed,

PROPOSITION 5.11
Let Δ be a finite set of propositional formulas and let $\mathcal{AF}(\Delta) = \langle Args(\Delta), Att \rangle$ be the argumentation framework that is induced by Δ as described above. If $Const(\Delta) = \{\langle \{\psi\}, \psi \rangle \mid \psi \in \Delta\}$ is p-admissible for $\mathcal{AF}(\Delta)$ then $\mathcal{CAF}(\Delta) = \langle Args(\Delta), Att, Const(\Delta) \rangle$ is a constrained argumentation framework and for every conflict tolerant semantics S of $\mathcal{CAF}(\Delta)$ it holds that $\Delta \mid\!\sim_{\mathsf{S}} \psi$ for every $\psi \in \Delta$.

[17] A similar approach for deductive argumentation in the context of defeasible reasoning goes back to [30]; We refer, e.g. to [14] for a comparison between the two approaches.

PROOF. Immediate from the definition of $\mathcal{CAF}(\Delta)$ and its semantics. ∎

We conclude this section with three further remarks:

(1) As noted previously, constraints may be useful for enforcing the acceptance of conflicting arguments (such as experimental results with contradictory conclusions, conflicting indications coming from equally reliable sources, etc). It is interesting to note, however, that if the set of constraints *is* conflict-free, so are the minimally conflicting p-complete extensions of the underlying CAF (and of course the other way around):

PROPOSITION 5.12
Let $\mathcal{CAF}$ be a constrained argumentation framework for an argumentation framework $\mathcal{AF}$ and a set of constraints *Const*. Then *Const* is conflict-free iff every minimally conflicting p-complete extension of $\mathcal{CAF}$ is conflict-free.

PROOF. If a minimally conflicting p-complete extension of $\mathcal{CAF}$ is conflict-free, than since it contains the set of constraints *Const*, the latter must be conflict-free as well. Conversely, if *Const* is conflict-free, than since it is also p-admissible, it is in particular admissible, and so it is extendable to a complete extension *Ext* of $\mathcal{AF}$. Now, *Ext* is a conflict-free p-complete extension of $\mathcal{CAF}$, and as such it is a minimally conflicting p-complete extension of $\mathcal{CAF}$. This also implies that any other minimally conflicting p-complete extension of $\mathcal{CAF}$ is conflict-free (otherwise it would not be minimally conflicting). ∎

(2) The constraints considered here are of a very basic form, and are given as a motivation for introducing conflict-tolerant semantics. Clearly, in reality more complex constraints may be needed, and in many cases this can be easily done in our framework (see Note 66 below), but this is beyond the scope of this article.

(3) For another example on how argumentation frameworks may be extended for incorporating constrains the reader is referred to [21]. The main difference is that in [21] conflict freeness is assumed, and so neither of the constraints nor the extensions of the framework may be contradictory. This assumption implies, in particular, that extensions may not be available for some constrained frameworks or may be empty. Recall that by Proposition 5.4 this is not possible in our case.

6 Representation of conflict-tolerant argumentation

In this section we provide a simple approach, based on propositional languages and quantifications over propositional variables, for representing the above mentioned conflict-tolerant argumentation semantics by a unified logical theory. We shall use a propositional language $\mathcal{L}$, consisting of a set of atomic formulas $\mathsf{Atoms}(\mathcal{L})$, the propositional constants t and f, and the logical symbols $\neg, \wedge, \vee, \supset$. In what follows, we denote by the lower-case letters p, q, r atomic formulas of $\mathcal{L}$, the Greek letters ψ, ϕ denote formulas in $\mathcal{L}$, and the calligraphic letters $\mathcal{T}$, $\mathcal{S}$ denote sets of formulas in $\mathcal{L}$ (called *theories*). The set of all atoms occurring in a formula ψ is denoted by $\mathsf{Atoms}(\psi)$, and the set of all the atoms occurring in a theory $\mathcal{T}$ is denoted by $\mathsf{Atoms}(\mathcal{T})$, i.e. $\mathsf{Atoms}(\mathcal{T}) = \bigcup_{\psi \in \mathcal{T}} \mathsf{Atoms}(\psi)$.

The formalism described in what follows is based on the idea that the four-valued signed systems used in [7] for representing conflict-free semantics of argumentation frameworks can be incorporated also for representing the conflict-tolerant semantics defined above. In the following sections, we describe how this can be done.

6.1 Four-valued semantics and signed formulas

The resemblance of our setting to Belnap's well-known four-valued framework for computerized reasoning [12] is evident. This framework also consists of four basic elements ('truth values'), two of them, denoted t for 'truth' and f for 'falsity', represent the classical truth assignments, and the other two, denoted $\bot$ and $\top$, intuitively represent lack of information and contradictory information (respectively) about the underlying assertions. As in our case, two values t and $\top$ (called the 'designated elements') are used for designating acceptable assertions (see also [6][18]).

The four elements mentioned above may be arranged in a lattice structure in which f is the minimal element, t is the maximal one, and the other two values are intermediate elements that are incomparable. The corresponding structure $\mathcal{FOUR}=(\{t,f,\top,\bot\},\leq)$ is a distributive lattice with an order reversing involution $\neg$, for which $\neg t=f$, $\neg f=t$, $\neg\top=\top$ and $\neg\bot=\bot$. We shall denote the meet and the join of this lattice by $\wedge$ and $\vee$, respectively. Another operator on $\mathcal{FOUR}$ which will be useful in the sequel is defined as follows: $a\supset b=t$ if $a\in\{f,\bot\}$, and $a\supset b=b$ otherwise. The truth tables of the basic connectives of $\mathcal{FOUR}$ are given below.[19]

$\vee$	t	f	$\top$	$\bot$
t	t	t	t	t
f	t	f	$\top$	$\bot$
$\top$	t	$\top$	$\top$	t
$\bot$	t	$\bot$	t	$\bot$

$\wedge$	t	f	$\top$	$\bot$
t	t	f	$\top$	$\bot$
f	f	f	f	f
$\top$	$\top$	f	$\top$	f
$\bot$	$\bot$	f	f	$\bot$

$\supset$	t	f	$\top$	$\bot$
t	t	f	$\top$	$\bot$
f	t	t	t	t
$\top$	t	f	$\top$	$\bot$
$\bot$	t	t	t	t

$\neg$	
t	f
f	t
$\top$	$\top$
$\bot$	$\bot$

Now, a *valuation* ν is a function that assigns to each atomic formula a truth value from $\{t,f,\bot,\top\}$, and $\nu(\mathsf{t})=t$, $\nu(\mathsf{f})=f$. Any valuation is extended to complex formulas in the obvious way, using the truth tables of the basic lattice connectives given above: $\nu(\neg\psi)=\neg\nu(\psi)$ and $\nu(\psi\circ\phi)=\nu(\psi)\circ\nu(\phi)$ for every $\circ\in\{\wedge,\vee,\supset\}$. A valuation ν *satisfies* ψ iff $\nu(\psi)\in\{t,\top\}$. A valuation that satisfies every formula in $\mathcal{T}$ is a *model* of $\mathcal{T}$. The set of models of $\mathcal{T}$ is denoted by $mod(\mathcal{T})$.

The four truth values may also be represented by pairs of two-valued components of the lattice $(\{0,1\},0<1)$ as follows: $t=(1,0)$, $f=(0,1)$, $\top=(1,1)$, $\bot=(0,0)$. The intuition behind this representation is that the the first component in the pair indicates whether the corresponding assertion should be accepted, while the second component indicates whether the assertion should be rejected (this, for instance, the value $(1,1)$ is associated with contradictory evidence). In terms of the pairwise representation, the basic operators of $\mathcal{FOUR}$ are representable as follows: For $x_1,x_2,y_1,y_2\in\{0,1\}$,

$$(x_1,y_1)\vee(x_2,y_2)=(x_1\vee x_2,\,y_1\wedge y_2),$$
$$(x_1,y_1)\wedge(x_2,y_2)=(x_1\wedge x_2,\,y_1\vee y_2),$$
$$(x_1,y_1)\supset(x_2,y_2)=(\neg x_1\vee x_2,\,x_1\wedge y_2),$$
$$\neg(x,y)=(y,x).$$

The representation of truth values in terms of pairs of two-valued components implies a similar way of representing four-valued valuations. A four-valued valuation ν may be represented in terms of a pair of two-valued components (ν_1,ν_2) by $\nu(p)=(\nu_1(p),\nu_2(p))$. So if, for instance, $\nu(p)=t$, then $\nu_1(p)=1$ and $\nu_2(p)=0$. Note also that $\nu=(\nu_1,\nu_2)$ is a four-valued model of $\mathcal{T}$ iff $\nu_1(\psi)=1$ for every $\psi\in\mathcal{T}$.

[18] Note that as our purpose is to represent and reason with situations in which conflicting arguments may be accepted, logics that are trivialized in the presence of contradictions are not adequate for our goal, thus non-classical logics are indeed necessary here.

[19] We refer to [6, 12] for further discussions on $\mathcal{FOUR}$ and the logics that are induced by this structure.

DEFINITION 6.1
A *signed alphabet* $\mathsf{Atoms}^{\pm}(\mathcal{L})$ is a set that consists of two symbols $p^{\oplus}, p^{\ominus}$ for each atom $p \in \mathsf{Atoms}(\mathcal{L})$. The language over $\mathsf{Atoms}^{\pm}(\mathcal{L})$ is denoted by $\mathcal{L}^{\pm}$. Now,

- The two-valued valuation ν^2 on $\mathsf{Atoms}^{\pm}(\mathcal{L})$ that is *induced by* (or *associated with*) a four-valued valuation $\nu^4 = (\nu_1, \nu_2)$ on $\mathsf{Atoms}(\mathcal{L})$, interprets $p^{\oplus}$ as $\nu_1(p)$ and $p^{\ominus}$ as $\nu_2(p)$.
- The four-valued valuation ν^4 on $\mathsf{Atoms}(\mathcal{L})$ that is *induced by* a two-valued valuation ν^2 on $\mathsf{Atoms}^{\pm}(\mathcal{L})$ is defined, for every atom $p \in \mathsf{Atoms}(\mathcal{L})$, by $\nu^4(p) = (\nu^2(p^{\oplus}), \nu^2(p^{\ominus}))$.

In what follows we denote by ν^2 a valuation into $\{0,1\}$, and by ν^4 a valuation into $\{t, f, \top, \bot\}$.

DEFINITION 6.2
For an atom p and formulas ψ, ϕ, we define the following formulas in $\mathcal{L}^{\pm}$:

$$
\begin{aligned}
&\tau_1(p) = p^{\oplus}, && \tau_2(p) = p^{\ominus},\\
&\tau_1(\neg\psi) = \tau_2(\psi), && \tau_2(\neg\psi) = \tau_1(\psi),\\
&\tau_1(\psi \wedge \phi) = \tau_1(\psi) \wedge \tau_1(\phi), && \tau_2(\psi \wedge \phi) = \tau_2(\psi) \vee \tau_2(\phi),\\
&\tau_1(\psi \vee \phi) = \tau_1(\psi) \vee \tau_1(\phi), && \tau_2(\psi \vee \phi) = \tau_2(\psi) \wedge \tau_2(\phi),\\
&\tau_1(\psi \supset \phi) = \neg\tau_1(\psi) \vee \tau_1(\phi), && \tau_2(\psi \supset \phi) = \tau_1(\psi) \wedge \tau_2(\phi).
\end{aligned}
$$

Given a set $\mathcal{T}$ of formulas in $\mathcal{L}$, we denote $\tau_i(\mathcal{T}) = \{\tau_i(\psi) \mid \psi \in \mathcal{T}\}$, for $i = 1, 2$.

We call $\tau_i(\psi)$ $(i = 1, 2)$ the *signed formulas* that are obtained from ψ. Next we recall some basic properties of signed formulas (see [3, 8] for the proofs).

PROPOSITION 6.3
If ν^2 is induced by ν^4 or ν^4 is induced by ν^2, then ν^4 satisfies a formula ψ iff ν^2 satisfies $\tau_1(\psi)$, and ν^4 satisfies $\neg\psi$ iff ν^2 satisfies $\tau_2(\psi)$.

DEFINITION 6.4
For a formula ψ in $\mathcal{L}$ we define the following signed formulas in $\mathcal{L}^{\pm}$:

$$
\begin{aligned}
&\mathsf{val}(\psi, t) = \tau_1(\psi) \wedge \neg\tau_2(\psi), && \mathsf{val}(\psi, f) = \neg\tau_1(\psi) \wedge \tau_2(\psi),\\
&\mathsf{val}(\psi, \top) = \tau_1(\psi) \wedge \tau_2(\psi), && \mathsf{val}(\psi, \bot) = \neg\tau_1(\psi) \wedge \neg\tau_2(\psi).
\end{aligned}
$$

PROPOSITION 6.5
If ν^2 is induced by ν^4, or ν^4 is induced by ν^2, then for every formula ψ, $\nu^4(\psi) = x$ iff $\nu^2(\mathsf{val}(\psi, x)) = 1$.

Note that by the last proposition there is a one-to-one correspondence between the four-valued models of $\mathcal{T}$ and the two-valued models of $\tau_1(\mathcal{T})$: ν^4 is a model of $\mathcal{T}$ if the two-valued valuation that is associated with ν^4 is a model of $\tau_1(\mathcal{T})$, and ν^2 is a model of $\tau_1(\mathcal{T})$ if the four-valued valuation that is associated with ν^2 is a model of $\mathcal{T}$.

6.2 *Signed theories for representing conflict-tolerant semantics*

Let us first represent p-admissible extensions (alternatively, labelings) and p-complete extensions (labelings) by signed theories, interpreted by four-valued semantics. As noted previously, we shall do this by extending the framework for formalizing conflict-free semantics, described in [7], using the results in Section 3.

First, we represent p-admissible extensions. As Proposition 3.11 indicates, p-admissible extensions are represented by a four-valued semantics, in which the labels in, out, none and both correspond, respectively, to the truth values t, f, $\bot$ and $\top$. Next, we formalize this.

DEFINITION 6.6
Given an argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$, we let $\mathsf{pADM}_{\mathcal{AF}}(x)$ be the following set of expressions:

$$\left\{ \begin{array}{l} \mathsf{val}(x,t) \supset \bigwedge_{y \in Args} \Big(\mathsf{att}(y,x) \supset \mathsf{val}(y,f)\Big), \\ \mathsf{val}(x,f) \supset \bigvee_{y \in Args} \Big(\mathsf{att}(y,x) \wedge \Big(\mathsf{val}(y,t) \vee \mathsf{val}(y,\top)\Big)\Big), \\ \mathsf{val}(x,\top) \supset \Big(\bigwedge_{y \in Args} \Big(\mathsf{att}(y,x) \supset \Big(\mathsf{val}(y,f) \vee \mathsf{val}(y,\top)\Big)\Big) \wedge \bigvee_{y \in Args} \Big(\mathsf{att}(y,x) \wedge \mathsf{val}(y,\top)\Big)\Big), \\ \mathsf{val}(x,\bot) \supset \bigwedge_{y \in Args} \Big(\mathsf{att}(y,x) \supset \Big(\mathsf{val}(y,f) \vee \mathsf{val}(y,\bot)\Big)\Big) \end{array} \right\}.$$

In the expressions defined above, x is a variable (to be sequentially substituted by the elements of $Args$), $\mathsf{val}(x,v)$ are the signed formulas in Definition 6.4, $\mathsf{att}(y,x)$ is replaced by the propositional constant t if $(y,x) \in Att$ (i.e. if y attacks x in $\mathcal{AF}$), and otherwise $\mathsf{att}(y,x)$ is replaced by the propositional constant f.

DEFINITION 6.7
Given an argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$, we denote by $\mathsf{pADM}_{\mathcal{AF}}[A_i/x]$ the expressions in Definition 6.6, evaluated with respect to the argument $A_i \in Args$. Also,

$$\mathsf{pADM}(\mathcal{AF}) = \bigcup_{A_i \in Args} \mathsf{pADM}_{\mathcal{AF}}[A_i/x].$$

The formulas in $\mathsf{pADM}_{\mathcal{AF}}[A_i/x]$ represent the application on A_i of the conditions listed in Definition 3.4, and which are represented in Definition 6.6. The theory $\mathsf{pADM}(\mathcal{AF})$ is therefore the application of those conditions to all the arguments of $\mathcal{AF}$.

EXAMPLE 6.8
Consider again the argumentation framework $\mathcal{AF}_1$ of Figure 1. In this case, $\mathsf{pADM}(\mathcal{AF}_1)$ is the following theory:

$\mathsf{val}(A,t) \supset \mathsf{val}(B,f)$,	$\mathsf{val}(A,f) \supset (\mathsf{val}(B,t) \vee \mathsf{val}(B,\top))$,
$\mathsf{val}(B,t) \supset \mathsf{val}(A,f)$,	$\mathsf{val}(B,f) \supset (\mathsf{val}(A,t) \vee \mathsf{val}(A,\top))$,
$\mathsf{val}(C,t) \supset \mathsf{val}(B,f)$,	$\mathsf{val}(C,f) \supset (\mathsf{val}(B,t) \vee \mathsf{val}(B,\top))$,
$\mathsf{val}(D,t) \supset \mathsf{val}(C,f)$,	$\mathsf{val}(D,f) \supset (\mathsf{val}(C,t) \vee \mathsf{val}(C,\top))$,
$\mathsf{val}(A,\top) \supset \mathsf{val}(B,\top)$,[20]	$\mathsf{val}(A,\bot) \supset (\mathsf{val}(B,f) \vee \mathsf{val}(B,\bot))$,
$\mathsf{val}(B,\top) \supset \mathsf{val}(A,\top)$,	$\mathsf{val}(B,\bot) \supset (\mathsf{val}(A,f) \vee \mathsf{val}(A,\bot))$,
$\mathsf{val}(C,\top) \supset \mathsf{val}(B,\top)$,	$\mathsf{val}(C,\bot) \supset (\mathsf{val}(B,f) \vee \mathsf{val}(B,\bot))$,
$\mathsf{val}(D,\top) \supset \mathsf{val}(C,\top)$,	$\mathsf{val}(D,\bot) \supset (\mathsf{val}(C,f) \vee \mathsf{val}(C,\bot))$

[20]This is a simplified formula of the original one, which is $\mathsf{val}(A,\top) \supset ((\mathsf{val}(B,f) \vee \mathsf{val}(B,\top)) \wedge \mathsf{val}(B,\top))$. We perform a similar rewriting on the other formulas in which $\mathsf{val}(x,\top)$ appears on the left-hand side of the implication.

More explicitly, in terms of signed propositional variables, $\mathsf{pADM}(\mathcal{AF}_1)$ is of the following form:

$$
\begin{array}{ll}
(A^{\oplus}\wedge\neg A^{\ominus})\supset(\neg B^{\oplus}\wedge B^{\ominus}), & (\neg A^{\oplus}\wedge A^{\ominus})\supset B^{\oplus},^{21} \\
(B^{\oplus}\wedge\neg B^{\ominus})\supset(\neg A^{\oplus}\wedge A^{\ominus}), & (\neg B^{\oplus}\wedge B^{\ominus})\supset A^{\oplus}, \\
(C^{\oplus}\wedge\neg C^{\ominus})\supset(\neg B^{\oplus}\wedge B^{\ominus}), & (\neg C^{\oplus}\wedge C^{\ominus})\supset B^{\oplus}, \\
(D^{\oplus}\wedge\neg D^{\ominus})\supset(\neg C^{\oplus}\wedge C^{\ominus}), & (\neg D^{\oplus}\wedge D^{\ominus})\supset C^{\oplus}, \\
(A^{\oplus}\wedge A^{\ominus})\supset(B^{\oplus}\wedge B^{\ominus}), & (\neg A^{\oplus}\wedge\neg A^{\ominus})\supset\neg B^{\oplus},^{22} \\
(B^{\oplus}\wedge B^{\ominus})\supset(A^{\oplus}\wedge A^{\ominus}), & (\neg B^{\oplus}\wedge\neg B^{\ominus})\supset\neg A^{\oplus}, \\
(C^{\oplus}\wedge C^{\ominus})\supset(B^{\oplus}\wedge B^{\ominus}), & (\neg C^{\oplus}\wedge\neg C^{\ominus})\supset\neg B^{\oplus}, \\
(D^{\oplus}\wedge D^{\ominus})\supset(C^{\oplus}\wedge C^{\ominus}), & (\neg D^{\oplus}\wedge\neg D^{\ominus})\supset\neg C^{\oplus}
\end{array}
$$

The (two-valued) models of the theory above are the following:

	$A^{\oplus}$	$A^{\ominus}$	$B^{\oplus}$	$B^{\ominus}$	$C^{\oplus}$	$C^{\ominus}$	$D^{\oplus}$	$D^{\ominus}$
μ_1	1	0	0	1	1	0	0	1
μ_2	1	0	0	1	0	0	0	0
μ_3	0	1	1	0	0	1	1	0
μ_4	0	1	1	0	0	1	0	0
μ_5	0	0	0	0	0	0	0	0

	$A^{\oplus}$	$A^{\ominus}$	$B^{\oplus}$	$B^{\ominus}$	$C^{\oplus}$	$C^{\ominus}$	$D^{\oplus}$	$D^{\ominus}$
μ_6	1	1	1	1	0	1	1	0
μ_7	1	1	1	1	0	1	0	0
μ_8	1	1	1	1	1	1	0	1
μ_9	1	1	1	1	1	1	1	1

The four-valued valuations that are associated with these models are the following:

	A	B	C	D
ν_1	t	f	t	f
ν_2	t	f	$\bot$	$\bot$
ν_3	f	t	f	t
ν_4	f	t	f	$\bot$
ν_5	$\bot$	$\bot$	$\bot$	$\bot$

	A	B	C	D
ν_6	$\top$	$\top$	f	t
ν_7	$\top$	$\top$	f	$\bot$
ν_8	$\top$	$\top$	$\top$	f
ν_9	$\top$	$\top$	$\top$	$\top$

The sets of atoms that are assigned values in $\{t,\top\}$ by these valuations are $\{A,C\}$, $\{A\}$, $\{B,D\}$, $\{B\}$, $\{\}$, $\{A,B,D\}$, $\{A,B\}$, $\{A,B,C\}$ and $\{A,B,C,D\}$. These are exactly the p-admissible extensions of $\mathcal{AF}_1$ (see Example 3.6), as indeed suggested by Corollary 6.12 below.

Note that here and in what follows we freely exchange an argument $A_i\in Args$, the propositional variable that represents A_i (with the same notation), and the corresponding signed variables $A_i^{\oplus}$, $A_i^{\ominus}$ in $\mathsf{pADM}(\mathcal{AF})$.

Given an argumentation framework $\mathcal{AF}=\langle Args,Att\rangle$ and a valuation ν on $Args^{\pm}$, we denote:

$$
\begin{aligned}
\mathsf{In}(\nu)&=\{A_i\in Args \mid \nu(A_i^{\oplus})=1,\ \nu(A_i^{\ominus})=0\},\\
\mathsf{Out}(\nu)&=\{A_i\in Args \mid \nu(A_i^{\oplus})=0,\ \nu(A_i^{\ominus})=1\},\\
\mathsf{Both}(\nu)&=\{A_i\in Args \mid \nu(A_i^{\oplus})=1,\ \nu(A_i^{\ominus})=1\}.\\
\mathsf{None}(\nu)&=\{A_i\in Args \mid \nu(A_i^{\oplus})=0,\ \nu(A_i^{\ominus})=0\}.
\end{aligned}
$$

[21] This is a simplified formula of the original one, which is $(A^{\ominus}\wedge\neg A^{\oplus})\supset((B^{\oplus}\wedge\neg B^{\ominus})\vee(B^{\oplus}\wedge B^{\ominus}))$. We perform a similar rewriting on other formulas of a similar form.

[22] Again, this is a simplified formula of the original one, which is $(\neg A^{\oplus}\wedge\neg A^{\ominus})\supset((B^{\ominus}\wedge\neg B^{\oplus})\vee(\neg B^{\ominus}\wedge\neg B^{\oplus}))$. We perform a similar rewriting on other formulas of a similar form.

NOTE 6.9
Given a two-valued valuation ν^2 on $Args^{\pm}$, let ν^4 be the four-valued valuation on $Args$ that is induced by ν^2 (Definition 6.1). Then $A \in \mathsf{In}(\nu^2)$ iff $\nu^4(A) = t$, $A \in \mathsf{Out}(\nu^2)$ iff $\nu^4(A) = f$, $A \in \mathsf{Both}(\nu^2)$ iff $\nu^4(A) = \top$, and $A \in \mathsf{None}(\nu^2)$ iff $\nu^4(A) = \bot$. Thus, if we associate ν^4 with a corresponding four-valued labeling $lab(\nu^4)$, defined by $lab(\nu^4)(A) = \mathsf{in}$ iff $\nu^4(A) = t$, $lab(\nu^4)(A) = \mathsf{out}$ iff $\nu^4(A) = f$, $lab(\nu^4)(A) = \mathsf{both}$ iff $\nu^4(A) = \top$, and $lab(\nu^4)(A) = \mathsf{none}$ iff $\nu^4(A) = \bot$, we have that: $A \in \mathsf{In}(\nu^2)$ iff $A \in \mathsf{In}(lab(\nu^4))$, $A \in \mathsf{Out}(\nu^2)$ iff $A \in \mathsf{Out}(lab(\nu^4))$, $A \in \mathsf{Both}(\nu^2)$ iff $A \in \mathsf{Both}(lab(\nu^4))$, and $A \in \mathsf{None}(\nu^2)$ iff $A \in \mathsf{None}(lab(\nu^4))$.

PROPOSITION 6.10
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Then for every p-admissible extension Ext of $\mathcal{AF}$ there is a model ν of $\mathsf{pADM}(\mathcal{AF})$, such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$ and $Ext^{+} = \mathsf{Out}(\nu) \cup \mathsf{Both}(\nu)$.

PROOF. Let Ext be a p-admissible extension of $\mathcal{AF}$. By Proposition 3.11, $p\mathcal{EL}_{\mathcal{AF}}(Ext)$ is a p-admissible labeling of $\mathcal{AF}$. Define now a valuation ν on $Args^{\pm}$ as follows:

$$\nu(A_i^{\oplus}) = \begin{cases} 1 & \text{if } p\mathcal{EL}(Ext)(A_i) \in \{\mathsf{in}, \mathsf{both}\}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\nu(A_i^{\ominus}) = \begin{cases} 1 & \text{if } p\mathcal{EL}(Ext)(A_i) \in \{\mathsf{out}, \mathsf{both}\}, \\ 0 & \text{otherwise.} \end{cases}$$

It holds that $\mathsf{In}(\nu) = \mathsf{In}(p\mathcal{EL}(Ext))$, $\mathsf{Out}(\nu) = \mathsf{Out}(p\mathcal{EL}(Ext))$, $\mathsf{None}(\nu) = \mathsf{None}(p\mathcal{EL}(Ext))$, and $\mathsf{Both}(\nu) = \mathsf{Both}(p\mathcal{EL}(Ext))$, therefore

$$Ext = \mathsf{In}(p\mathcal{EL}(Ext)) \cup \mathsf{Both}(p\mathcal{EL}(Ext)) = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$$

and

$$Ext^{+} = \mathsf{Out}(p\mathcal{EL}(Ext)) \cup \mathsf{Both}(p\mathcal{EL}(Ext)) = \mathsf{Out}(\nu) \cup \mathsf{Both}(\nu).$$

It remains to show that ν is a model of $\mathsf{pADM}(\mathcal{AF})$. Indeed, suppose for instance that $\nu(A^{\oplus}) = 0$ and $\nu(A^{\ominus}) = 1$ for some $A \in Args$ (the other three cases are similar). This means, in particular, that $\nu(\mathsf{val}(A,t)) = \nu(\mathsf{val}(A,\top)) = \nu(\mathsf{val}(A,\bot)) = 0$. Thus, ν satisfies the formulas that are obtained from the first, third and fourth expressions in Definition 6.6 for $x = A$. To see that ν also satisfies the second expression in that definition note that by our assumptions on ν and by its definition it holds that $p\mathcal{EL}(Ext)(A) = \mathsf{out}$. Now, since $p\mathcal{EL}(Ext)$ is a p-admissible labeling of $\mathcal{AF}$, it in particular satisfies **pOut**, and so there exists some $B \in A^{-}$ for which $p\mathcal{EL}(Ext)(B) \in \{\mathsf{in}, \mathsf{both}\}$. For this B we have that $\mathsf{att}(B,A)$ is replaced in the signed theory by the constant t and that $\nu(B^{\oplus}) = 1$, i.e. $B \in \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$. It follows that $\nu(\mathsf{val}(B,t) \vee \mathsf{val}(B,\top)) = 1$, and so

$$\nu\Big(\bigvee_{y \in Args} \Big(\mathsf{att}(y,A) \wedge \Big(\mathsf{val}(y,t) \vee \mathsf{val}(y,\top) \Big) \Big) \Big) = 1.$$

This implies that ν satisfies also the formula corresponding to the second expression in Definition 6.6 (for $x = A$). ∎

PROPOSITION 6.11
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Then for every model ν of $\mathsf{pADM}(\mathcal{AF})$ there is a p-admissible extension Ext of $\mathcal{AF}$ such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$.

PROOF. Let ν be a model of $\mathsf{pADM}(\mathcal{AF})$. Define a four-valued labeling lab_ν by $lab_\nu(A) = \mathsf{in}$ if $\nu(A) \in \mathsf{In}(\nu)$, $lab_\nu(A) = \mathsf{out}$ if $\nu(A) \in \mathsf{Out}(\nu)$, $lab_\nu(A) = \mathsf{both}$ if $\nu(A) \in \mathsf{Both}(\nu)$, and $lab_\nu(A) = \mathsf{none}$

if $\nu(A) \in \mathsf{None}(\nu)$. It is easy to verify that since ν is a model of $\mathsf{pADM}(\mathcal{AF})$, lab_ν satisfies the four conditions in Definition 3.4. For instance, to see **pIn**, suppose that $lab_\nu(A) = \mathsf{in}$ for some $A \in Args$. Then $\nu(A) \in \mathsf{In}(\nu)$, i.e. $\nu(A^{\oplus}) = 1$ and $\nu(A^{\ominus}) = 0$. Thus, $\mathsf{val}(A,t) = 1$. By the first expression of Definition 6.6 when $x = A$, then,

$$\nu\Big(\bigwedge_{y \in Args}\Big(\mathsf{att}(y,A) \supset \mathsf{val}(y,f)\Big)\Big) = 1,$$

which implies that for every attacker B of A, $\nu(\mathsf{val}(B,f)) = 1$. Hence, for such an attacker B, it holds that $B \in \mathsf{Out}(\nu)$, and so $lab_\nu(B) = \mathsf{out}$. Similarly, the other three expressions in Definition 6.6 guarantee conditions **pOut**, **pBoth** and **pNone** in Definition 3.4. It follows that lab_ν is a p-admissible labeling of $\mathcal{AF}$, and by Proposition 3.12 $Ext_\nu = p\mathcal{LE}(lab_\nu)$ is a p-admissible extension for $\mathcal{AF}$. Moreover, we have that $Ext_\nu = \mathsf{In}(lab_\nu) \cup \mathsf{Both}(lab_\nu) = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$. ■

By the last two propositions we have the following corollary.

Corollary 6.12
The set of the p-admissible extensions of an argumentation framework $\mathcal{AF}$ is the same as the set $\{\mathsf{In}(\nu) \cup \mathsf{Both}(\nu) \mid \nu \text{ is a model of } \mathsf{pADM}(\mathcal{AF})\}$.

Next, we represent p-complete extensions. Again, the idea is to formalize the conditions of such extensions (Definition 3.5) by a corresponding signed theory. Below, we abbreviate by $\psi \leftrightarrow \phi$ the formula $(\psi \supset \phi) \wedge (\phi \supset \psi)$.

Definition 6.13
Given an argumentation framework $\mathcal{AF} = \langle Args, Att \rangle$, we let $\mathsf{pCMP}_{\mathcal{AF}}(x)$ be the following set of expressions:

$$\left\{\begin{array}{l}
\mathsf{val}(x,t) \leftrightarrow \bigwedge_{y \in Args}\Big(\mathsf{att}(y,x) \supset \mathsf{val}(y,f)\Big),\\
\mathsf{val}(x,f) \leftrightarrow \Big(\bigvee_{y \in Args}\Big(\mathsf{att}(y,x) \wedge \Big(\mathsf{val}(y,t) \vee \mathsf{val}(y,\top)\Big)\Big) \wedge \bigvee_{y \in Args}\Big(\mathsf{att}(y,x) \wedge \Big(\mathsf{val}(y,t) \vee \mathsf{val}(y,\bot)\Big)\Big)\Big),\\
\mathsf{val}(x,\top) \leftrightarrow \Big(\bigwedge_{y \in Args}\Big(\mathsf{att}(y,x) \supset \Big(\mathsf{val}(y,f) \vee \mathsf{val}(y,\top)\Big)\Big) \wedge \bigvee_{y \in Args}\Big(\mathsf{att}(y,x) \wedge \mathsf{val}(y,\top)\Big)\Big),\\
\mathsf{val}(x,\bot) \leftrightarrow \Big(\bigwedge_{y \in Args}\Big(\mathsf{att}(y,x) \supset \Big(\mathsf{val}(y,f) \vee \mathsf{val}(y,\bot)\Big)\Big) \wedge \bigvee_{y \in Args}\Big(\mathsf{att}(y,x) \wedge \mathsf{val}(y,\bot)\Big)\Big)
\end{array}\right\}.$$

As before, $\mathsf{pCMP}_{\mathcal{AF}}[A_i/x]$ denotes the substitution in the expressions above of x by an argument $A_i \in Args$, and

$$\mathsf{pCMP}(\mathcal{AF}) = \bigcup_{A_i \in Args} \mathsf{pCMP}_{\mathcal{AF}}[A_i/x].$$

Once again, we show the correspondence between the models of $\mathsf{pCMP}(\mathcal{AF})$ and the p-complete extensions of $\mathcal{AF}$.

Proposition 6.14
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Then for every p-complete extension Ext of $\mathcal{AF}$ there is a model ν of $\mathsf{pCMP}(\mathcal{AF})$, such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$ and $Ext^{+} = \mathsf{Out}(\nu) \cup \mathsf{Both}(\nu)$.

Proposition 6.15
Let $\mathcal{AF} = \langle Args, Att \rangle$ be an argumentation framework. Then for every model ν of $\mathsf{pCMP}(\mathcal{AF})$ there is a p-complete extension Ext of $\mathcal{AF}$ such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$.

The proofs of Propositions 6.14 and 6.15 are similar to those of Propositions 6.10 and 6.11 (respectively), where the conditions of Definition 3.4 are replaced by the conditions of Definition 3.5.

COROLLARY 6.16
The set of the p-complete extensions of an argumentation framework $\mathcal{AF}$ is the same as the set $\{\mathsf{In}(\nu)\cup\mathsf{Both}(\nu)\mid \nu \text{ is a model of } \mathsf{pCMP}(\mathcal{AF})\}$.

EXAMPLE 6.17
The signed theory $\mathsf{pCMP}(\mathcal{AF}_1)$, where $\mathcal{AF}_1$ is the argumentation framework of Figure 1, is given below (using the abbreviations of Definition 6.4, and applying a few simple rewriting rules).

$$
\begin{array}{ll}
\mathsf{val}(A,t)\supset\mathsf{val}(B,f), & \mathsf{val}(A,f)\supset\mathsf{val}(B,t),\\
\mathsf{val}(B,t)\supset\mathsf{val}(A,f), & \mathsf{val}(B,f)\supset\mathsf{val}(A,t),\\
\mathsf{val}(C,t)\supset\mathsf{val}(B,f), & \mathsf{val}(C,f)\supset\mathsf{val}(B,t),\\
\mathsf{val}(D,t)\supset\mathsf{val}(C,f), & \mathsf{val}(D,f)\supset\mathsf{val}(C,t),\\
\\
\mathsf{val}(A,\top)\supset\mathsf{val}(B,\top), & \mathsf{val}(A,\bot)\supset\mathsf{val}(B,\bot),\\
\mathsf{val}(B,\top)\supset\mathsf{val}(A,\top), & \mathsf{val}(B,\bot)\supset\mathsf{val}(A,\bot),\\
\mathsf{val}(C,\top)\supset\mathsf{val}(B,\top), & \mathsf{val}(C,\bot)\supset\mathsf{val}(B,\bot),\\
\mathsf{val}(D,\top)\supset\mathsf{val}(C,\top), & \mathsf{val}(D,\bot)\supset\mathsf{val}(C,\bot)
\end{array}
$$

The four-valued valuations that are associated with the models of this theory are the following:

	A	B	C	D
ν_1	t	f	t	f
ν_2	f	t	f	t
ν_3	$\bot$	$\bot$	$\bot$	$\bot$
ν_4	$\top$	$\top$	$\top$	$\top$

The sets of atoms that are assigned values in $\{t,\top\}$ by these valuations are $\{A,C\}$, $\{B,D\}$, $\{\}$ and $\{A,B,C,D\}$. These are exactly the p-complete extensions of $\mathcal{AF}_1$ (see Example 3.6), as indeed guaranteed by Corollary 6.16.

6.2.1 Constraint handling

Handling constrains in argumentation systems is very simple. For forcing the inclusion of the arguments in a given set of constraints *Const*, we just have to make sure that their variables would get designated values by the models of the corresponding signed theory. This is assured by the following sets of formulas:

$$\Big\{\mathsf{val}(A,t)\vee\mathsf{val}(A,\top)\mid A\in Const\Big\}.$$

An equivalent (and somewhat more explicit and simplified) writing of the formulas above is by the formula

$$\mathsf{Const}(Args)=\bigwedge_{A\in Const} A^{\oplus}.$$

Let $\mathcal{CAF}=\langle\mathcal{AF},Const\rangle$ be a constrained argumentation framework, where $\mathcal{AF}=\langle Args,Att\rangle$. By Corollary 6.12, the p-admissible extensions of $\mathcal{CAF}$ are the sets $\mathsf{In}(\nu)\cup\mathsf{Both}(\nu)$, where ν ranges over the models of the following signed theory:

$$\mathsf{pADM}(\mathcal{CAF})=\mathsf{pADM}(\mathcal{AF})\cup\{\mathsf{Const}(Args)\}.$$

Similarly, by Corollary 6.16, the p-complete extensions of $\mathcal{CAF}$ are the sets $\mathsf{In}(\nu)\cup\mathsf{Both}(\nu)$, where ν ranges over the models of the following signed theory:

$$\mathsf{pCMP}(\mathcal{CAF})=\mathsf{pCMP}(\mathcal{AF})\cup\{\mathsf{Const}(Args)\}.$$

NOTE 6.18
By using Definition 6.2, it is easy to incorporate more complex forms of constraints. For instance, demanding that the acceptance of argument A implies the acceptance of argument B may be formalized by the introduction of the constraint $A\supset B$. This means the addition of the signed formula $\tau_1(A\supset B)=\neg A^{\oplus}\vee B^{\oplus}$ to the corresponding signed theory.

6.3 *Signed QBF theories for conflict minimization*

As indicated in Section 5, when the set of conflicts of a constrained argumentation framework is not conflict-free all of its p-admissible and p-complete extensions would contain conflicting arguments. In such cases, it may be useful to select only those extensions in which the number of conflicts are as minimal as possible (Definition 5.5). For computing those minimally conflicting p-extensions, we use the same approach as in [7] and incorporate quantified Boolean formulas (QBFs) [15] for formalizing conflicts minimizations.

Quantified Boolean formulas are obtained by extending the underlying propositional language $\mathcal{L}$ with universal and existential quantifiers $\forall,\exists$ over propositional variables. The intuitive meaning of a QBF of the form $\exists p\forall q\,\psi$, for instance, is that there exists a truth assignment of p such that for every truth assignment of q, ψ is true. Clearly, every QBF is associated with a logically equivalent propositional formula, thus QBFs can be seen as a conservative extension of classical propositional logic.

DEFINITION 6.19
Let Ψ be a QBF and Γ a set of QBFs.

- An occurrence of an atom p in Ψ is called *free*, if it is not in the scope of a quantifier $\mathsf{Q}p$, for $\mathsf{Q}\in\{\forall,\exists\}$. We denote by $\Psi[\phi_1/p_1,\dots,\phi_n/p_n]$ the uniform substitution of each free occurrence of a variable (atom) p_i in Ψ by a formula ϕ_i, for $i=1,\dots,n$.
- The definition of a *valuation* can be extended to QBFs as follows:

$$\begin{aligned}
&\nu(\neg\psi)=\neg\nu(\psi),\\
&\nu(\psi\circ\phi)=\nu(\psi)\circ\nu(\phi),\ \text{where}\ \circ\in\{\wedge,\vee,\supset\},\\
&\nu(\forall p\,\psi)=\nu(\psi[\mathsf{t}/p])\wedge\nu(\psi[\mathsf{f}/p]),\\
&\nu(\exists p\,\psi)=\nu(\psi[\mathsf{t}/p])\vee\nu(\psi[\mathsf{f}/p]).
\end{aligned}$$

- We say that a (two-valued) valuation ν *satisfies* Ψ if $\nu(\Psi)=1$. A valuation ν is a *model* of Γ if ν satisfies every element of Γ. We say that Ψ is (classically) *entailed by* Γ, if every model of Γ is also a model of Ψ.

In order to compute the p-admissible extensions of a constrained argumentation framework $\mathcal{CAF}$, we should identify the models of $\mathsf{pADM}(\mathcal{CAF})$ and exclude those whose set of $\top$-assignments is not minimal with respect to set inclusion. This is what we do by the circumscriptive-like QBF that is defined next.

Definition 6.20
Given a constrained argumentation theory $\mathcal{CAF} = \langle Args, Att, Const\rangle$ in which $|Args| = n$, let $\mathsf{pADM}(\mathcal{CAF})$ be the signed theory for computing the p-admissible extensions of $\mathcal{CAF}$ defined in the previous section. Let also $Args^{\pm} = \{A_i^{\oplus} \mid A_i \in Ar\} \cup \{A_i^{\ominus} \mid A_i \in Ar\}$ be the set of atoms in $\mathsf{pADM}(\mathcal{CAF})$. We denote by $\bigwedge \mathsf{pADM}(\mathcal{CAF})$ the conjunction of the formulas in $\mathsf{pADM}(\mathcal{CAF})$. Now, we denote by $\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))$ the following QBF:

$$\forall p_1^{\oplus}, p_1^{\ominus}, \ldots, p_n^{\oplus}, p_n^{\ominus} \Big(\bigwedge \mathsf{pADM}(\mathcal{CAF}) \Big[p_1^{\oplus}/A_1^{\oplus}, p_1^{\ominus}/A_1^{\ominus}, \ldots, p_n^{\oplus}/A_n^{\oplus}, p_n^{\ominus}/A_n^{\ominus} \Big] \supset$$
$$\Big(\bigwedge_{A_i \in Args} \Big(\mathsf{val}(A_i, \top) \Big[p_i^{\oplus}/A_i^{\oplus}, p_i^{\ominus}/A_i^{\ominus} \Big] \supset \mathsf{val}(A_i, \top) \Big) \supset$$
$$\bigwedge_{A_i \in Args} \Big(\mathsf{val}(A_i, \top) \supset \mathsf{val}(A_i, \top) \Big[p_i^{\oplus}/A_i^{\oplus}, p_i^{\ominus}/A_i^{\ominus} \Big] \Big) \Big) \Big).$$

As we shall see shortly, among the models of $\mathsf{pADM}(\mathcal{CAF})$, the only ones who satisfy the formula above are those with minimal $\top$-assignments (where minimization here is with respect to set inclusion; cf. Definition 5.5). This brings us to the next definition.

Definition 6.21
Given a constrained argumentation theory $\mathcal{CAF}$, we denote

$$\mathsf{MINpADM}(\mathcal{CAF}) = \mathsf{pADM}(\mathcal{CAF}) \cup \{\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))\}.$$

Proposition 6.22
Let $\mathcal{CAF} = \langle Args, Att\rangle$ be a constrained argumentation framework. A subset Ext of $Args$ is a minimally conflicting p-admissible extension of $\mathcal{CAF}$ iff there is a model ν of $\mathsf{MINpADM}(\mathcal{CAF})$ such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$.

Proof. By Corollary 6.12 it only remains to show that ν is a model of $\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))$ iff there is no model μ of $\mathsf{pADM}(\mathcal{CAF})$ for which $\mathsf{Both}(\mu) \subsetneq \mathsf{Both}(\nu)$. Indeed, by Definition 6.20, and since for every $A_i \in Args$ it holds that $\nu(\mathsf{val}(A_i, \top)) = 1$ iff $A_i \in \mathsf{Both}(\nu)$, we have that ν is a model of $\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))$ iff for every model μ of $\mathsf{pADM}(\mathcal{CAF})$ such that $\mathsf{Both}(\mu) \subseteq \mathsf{Both}(\nu)$, also $\mathsf{Both}(\nu) \subseteq \mathsf{Both}(\mu)$. Thus, ν satisfies $\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))$ iff $\mathsf{Both}(\nu)$ is not properly contained in any set of the form $\mathsf{Both}(\mu)$ for some model μ of $\mathsf{pADM}(\mathcal{CAF})$. ■

The p-complete extensions of a constrained argumentation theory $\mathcal{CAF}$ are computed similarly. Let $\mathsf{Min}_{\top}(\mathsf{pCMP}(\mathcal{CAF}))$ be a signed QBF that is similar to the signed QBF $\mathsf{Min}_{\top}(\mathsf{pADM}(\mathcal{CAF}))$ in Definition 6.20, except that $\bigwedge \mathsf{pADM}(\mathcal{CAF})$ is replaced by the conjunction $\bigwedge \mathsf{pCMP}(\mathcal{CAF})$ of the formulas in $\mathsf{pCMP}(\mathcal{CAF})$. We denote:

$$\mathsf{MINpCMP}(\mathcal{CAF}) = \mathsf{pCMP}(\mathcal{CAF}) \cup \{\mathsf{Min}_{\top}(\mathsf{pCMP}(\mathcal{CAF}))\}.$$

Then we have the following proposition, the proof of which is similar to that of Proposition 6.22.

Proposition 6.23
Let $\mathcal{CAF} = \langle Args, Att\rangle$ be a constrained argumentation framework. A subset Ext of $Args$ is a minimally conflicting p-complete extension of $\mathcal{CAF}$ iff there is a model ν of $\mathsf{MINpCMP}(\mathcal{CAF})$ such that $Ext = \mathsf{In}(\nu) \cup \mathsf{Both}(\nu)$.

NOTE 6.24

Like minimally conflicting extensions, p-semi-stable p-admissible extensions (Definition 5.7) may be computed by augmenting the theory $\mathsf{MINpADM}(\mathcal{CAF})$ by the following signed QBF, assuring minimal $\perp$-assignments among the minimally conflicting p-admissible extensions of $\mathcal{CAF}$:

$$\forall p_1^{\oplus}, p_1^{\ominus}, \ldots, p_n^{\oplus}, p_n^{\ominus} \Big(\bigwedge \mathsf{MINpADM}(\mathcal{CAF}) \Big[p_1^{\oplus}/A_1^{\oplus}, p_1^{\ominus}/A_1^{\ominus}, \ldots, p_n^{\oplus}/A_n^{\oplus}, p_n^{\ominus}/A_n^{\ominus} \Big] \supset \Big(\bigwedge_{A_i \in Args} \Big(\mathsf{val}(A_i, \perp) \Big[p_i^{\oplus}/A_i^{\oplus}, p_i^{\ominus}/A_i^{\ominus} \Big] \supset \mathsf{val}(A_i, \perp) \Big) \supset \bigwedge_{A_i \in Args} \Big(\mathsf{val}(A_i, \perp) \supset \mathsf{val}(A_i, \perp) \Big[p_i^{\oplus}/A_i^{\oplus}, p_i^{\ominus}/A_i^{\ominus} \Big] \Big) \Big) \Big).$$

For having the p-semi-stable p-complete extensions of $\mathcal{CAF}$, we have to augment the theory $\mathsf{MINpCMP}(\mathcal{CAF})$ be a QBF that is similar to the one above, where $\bigwedge \mathsf{MINpADM}(\mathcal{CAF})$ is replaced by $\bigwedge \mathsf{MINpCMP}(\mathcal{CAF})$.

7 Conclusion

The lack of satisfactory facilities for dealing with arguments that, directly or indirectly, contradict themselves is already indicated in [16] and [22]. This issue has attracted a considerable attention in recent years and several argumentation semantics were proposed in order to properly maintain loop situations. In this article, we considered a clement approach to circularity in argumentation frameworks, derived by four-valued labelings and corresponding extensions that may not be conflict-free. Our conflict-tolerant approach to abstract argumentation is beneficial for several reasons:

- From a purely theoretical point of view, we have shown that the correlation between the labeling-based approach and the extension-based approach to argumentation theory is preserved also when conflict-freeness is abandoned. Interestingly, as indicated in Note 3.16, in our framework this correlation holds also between admissibility-based labelings and admissibility-based extensions, which is *not* the case in the conflict-free setting of [19].
- From a more pragmatic point of view, new types of semantics are introduced, which accommodate conflicts, yet they are not trivialized by inconsistency. It is shown that this setting is not a substitute of standard (conflict-free) semantics, but rather a generalized framework, offering an option for inter-attacks when such attacks make sense or are unavoidable.
- As demonstrated in Section 5, in some extended forms of argumentation frameworks conflicts among accepted arguments cannot be avoided. This could be the case, for instance, when constraints are introduced. In such contexts the necessity of maintaining conflicts is evident.
- Already in standard approaches for giving semantics to argumentation systems the issue of conflicts handling turns out to be more evasive than what it looks like at first sight. In fact, conflicts may implicitly arise even in conflict-free semantics, because such semantics simulate binary attacks and not collective conflicts. To see this, consider the last example of [9]: ‘John will be on the tandem bicycle because he wants to’, ‘Mary will be on the tandem bicycle because she wants to’ and ‘Suzy will be on the tandem bicycle because she wants to’. These three arguments are in collective conflict when the tandem has only two seats. Indeed, as noted in [9], conflict-freeness without admissibility is not enough for guaranteeing consistent conclusions. In this respect, the possibility of having conflicts is not completely ruled out even

in some conflict-free semantics (such as CF2 and stage semantics; see [9]), and our approach may be viewed as an explication of this possibility.

As we have shown, our conflict-tolerant approach to abstract argumentation theory may be represented in terms of a logical theory, based on signed formulas. Such a theory can serve as the basis for representing and computing various decision problems involving contradictory arguments. This purely logical approach makes problems like sceptical and credulous acceptance of arguments simply a matter of entailment and satisfiability checking. The latter may be verified by off-the-shelf SAT-solvers and QBF-solvers.

Finally, it would be interesting, and probably helpful, to introduce evaluation criteria for conflict-tolerant semantics, similar to those considered in [10] for conflict-free semantics. This remains a subject for future work.

References

[1] L. Amgoud and P. Besnard. Bridging the gap between abstract argumentation systems and logic. In *Proceedings of 3rd International Conference on Scalable Uncertainty Management (SUM'09)*, vol. 5785 of *Lecture Notes in Computer Science*, pp. 12–27. Springer, 2009.

[2] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, **29**, 125–169, 2002.

[3] O. Arieli. Paraconsistent reasoning and preferential entailments by signed quantified Boolean formulas. *ACM Transactions on Computational Logic*, **8**, Article 18, 2007.

[4] O. Arieli. Conflict-tolerant semantics for argumentation frameworks. In *Proceedings of 13th European Conference on Logics in Artificial Intelligence (JELIA'12)*, vol. 7519 of *Lecture Notes in Computer Science*, L. Fariñas del Cerro, A. Herzig, and J. Mengin, eds, pp. 28–40. Springer, 2012.

[5] O. Arieli. Towards constraints handling by conflict tolerance in abstract argumentation arameworks. In *Proceedings of 26th International FLAIRS Conference (Special Track on Uncertain Reasoning)*, C. Boonthem-Denecke and G. M. Youngblood, eds, pp. 585–590. AAAI Press, 2013.

[6] O. Arieli and A. Avron. The value of the four values. *Artificial Intelligence*, **102**, 97–141, 1998.

[7] O. Arieli and M. W. A. Caminada. A general QBF-based framework for formalizing argumentation. In *Proceedings of 4th Conference on Computational Models of Argument (COMMA'12)*, vol. 245 of *Frontiers in Artificial Intelligence and Applications*, pp. 105–116. IOS Press, 2012.

[8] O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, **13**, 557–580, 2003.

[9] P. Baroni, M. W. A. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, **26**, 365–410, 2011.

[10] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, **171**, 675–700, 2007.

[11] P. Baroni and M. Giacomin. Semantics for abstract argumentation systems. In *Argumentation in Artificial Intelligence*, I. Rahwan and G. R. Simari, eds, pp. 25–44. Springer, 2009.

[12] N. D. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logics*, J. M. Dunn and G. Epstein, eds, pp. 7–37. Reidel Publishing Company, 1977.

[13] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, **128**, 203–235, 2001.

[14] Ph. Besnard and A. Hunter. Argumentation based on classical logic. In *Argumentation in Artificial Intelligence*, I. Rahwan and G. R. Simari, eds, pp. 133–152. Springer, 2009.

[15] Ph. Besnard, T. Schaub, H. Tompits, and S. Woltran. Representing paraconsistent reasoning via quantified propositional logic. In *Inconsistency Tolerance*, number 3300 in *Lecture Notes in Computer Science*, L. Bertossi, A. Hunter, and T. Schaub, eds, pp. 84–118. Springer, 2004.

[16] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, **93**, 63–101, 1997.

[17] M. W. A. Caminada. On the issue of reinstatement in argumentation. In *Proceedings of 10th European Conference on Logics in Artificial Intelligence (JELIA'06)*, vol. 4160 of *Lecture Notes in Computer Science*, M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, eds, pp. 111–123. Springer, 2006.

[18] M. W. A. Caminada, W. A. Carnielli, and P. E. Dunne. Semi-stable semantics. *Journal of Logic and Computation*, **22**, 1207–1254, 2012.

[19] M. W. A. Caminada and D. M. Gabbay. A logical account of formal argumentation. *Studia Logica*, **93**, 109–145, 2009.

[20] S. Coste-Marquis, C. Devred, and P. Marquis. Prudent semantics for argumentation frameworks. In *Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pp. 568–572. IEEE Computer Society, 2005.

[21] S. Coste-Marquis, C. Devred, and P. Marquis. Constrained argumentation frameworks. In *Proceedings of International Conference on the principles of knowledge representation and reasoning (KR'06)*, pp. 112–122. AAAI Press, 2006.

[22] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence*, **77**, 321–357, 1995.

[23] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, **171**, 642–674, 2007.

[24] W. Dvořák. On the complexity of computing the justification status of an argument. In *Post proc. 1st International Workshop on the Theory and Applications of Formal Argumentation (TAFA-11)*, vol. 7132 of *Lecture Notes in Atrificial Intelligence*, pp. 32–49. Springer, 2011.

[25] U. Egly, S. A. Gaggl, and S. Woltran. Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, **1**, 144–177, 2010.

[26] H. Jakobovits and D. Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, **9**, 215–261, 1999.

[27] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, **173**, 901–934, 2009.

[28] J. L. Pollock. Justification and defeat. *Artificial Intelligence*, **67**, 377–407, 1994.

[29] I. Rahwan and G. R. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

[30] G. R. Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, **53**, 125–157, 1992.

[31] A. Tarski. *Introduction to Logic*. Oxford University Press, 1941.

[32] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In *Proceedings of 8th Dutch Conference on Artificial Intelligence (NAIC'96)*, J. J. Ch. Meyer and L. C. van der Gaag, eds, pp. 357–368, 1996.

Received 26 March 2013