Conflict-Free and Conflict-Tolerant Semantics for Constrained Argumentation Frameworks

Ofer Arieli

School of Computer Science, The Academic College of Tel-Aviv, Israel

Abstract

In this paper we incorporate integrity constraints in abstract argumentation frameworks. Two types of semantics are considered for these constrained frameworks: conflict-free and conflict-tolerant. The first one is a conservative extension of standard approaches for giving coherent-based semantics to argumentation frameworks, where in addition certain constraints must be satisfied. A primary consideration behind this approach is a dismissal of any contradiction between accepted arguments of the constrained frameworks. The second type of semantics preserves contradictions, which are regarded as meaningful and sometimes even critical for the conclusions. We show that this approach is particularly useful for assuring the existence of non-empty extensions and for handling contradictions among the constraints, in which cases conflict-free extensions are not available.

Both types of semantics are represented by propositional sets of formulas and are evaluated in the context of three-valued and four-valued logics. Among others, we show a one-to-one correspondence between the models of these theories, the extensions, and the labelings of the underlying constrained argumentation frameworks.

1. Introduction

Dung's argumentation framework [1] is a graph-style representation of what may be viewed as a dispute. It is instantiated by a set of abstract objects, called arguments, and a binary relation on this set that intuitively represents attacks between arguments. These structures have been found useful for modeling a range of formalisms for non-monotonic reasoning, including default logic [2], logic programming under stable model semantics [3], three-valued stable model semantics [4] and well-founded model semantics [5], Nute's defeasible logic [6], and so on.

Despite of their general nature, experience shows that in some cases argumentation frameworks lack sufficient expressivity for accurately capturing their

Email address: oarieli@mta.ac.il (Ofer Arieli)

Preprint submitted to Elsevier *September 18, 2014*

domain, and some extra apparatus is needed to gain a more comprehensive representation of the relations among the arguments. This observation motivated several works, like those of Amgoud and Cayrol [7], Bench-Capon [8], and Modgil [9], in which in addition to the argumentation framework itself, some additional meta-knowledge is provided, e.g. in terms of ranking values or preference relations on the arguments. This helps to refine and improve the process of selecting the arguments that can collectively be accepted according to the argumentation framework at hand.

In this paper we formalize the additional knowledge that is linked to argumentation frameworks in terms of *integrity constraints*, that is, conditions that every accepted set of arguments must satisfy. Let us demonstrate the advantages of using constraints by means of a few simple and (for the time being) informal examples.

Example 1. Medical systems, as well as legal systems, are rule-based, and as such they are naturally representable by argumentation frameworks (see, e.g., [10]). Yet, even in these systems not all the rules are of equal importance or relevance for specific cases. Thus, for instance, arguments referring to concrete results concerning medical tests of a particular patient are usually given precedence over, say, arguments referring to general symptoms of a disease. This may be expressed by constraints obliging the reasoner to take these test results into account when stating a diagnosis (i.e., include them in every accepted set of arguments obtained by the framework), or by extra rules that confront arguments that not necessarily attack one another. More generally, integrity constraints provide means of expressing relations among the arguments which are not representable by 'standard' attack relations.

Example 2. The incorporation of constraints may be useful in handling scenarios where an argumentation framework is viewed as a dynamic process [11]. For instance, constraints may encode the *expected outcome* of an argumentation framework, or may help to evaluate the consequences of an argumentation framework in light of new arguments (see [12]).

Example 3. Constraints may also serve as a means for keeping the semantics of an argumentation framework coherent. To see this, consider the three arguments in the last example of [13]: "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". Here, integrity constraints may explicitly specify that these three arguments are in a collective conflict when the tandem has only two seats – a fact which is difficult to grasp only by standard semantical approaches to argumentation systems (see [13]).

Example 4. The use of meta-knowledge, e.g. in terms of integrity constraints, is a convenient way for accommodating conflicting arguments. Consider, for instance, an information system representing information about the theory of light. Here, 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. The incorporation of suitable constraints, forcing the acceptance of *both* arguments, could be an effective way of keeping the underlying theory realistic and non-biased.

Interestingly, in the last two examples integrity constraints have opposite roles: in Example 3 (and often also in the context of Example 2) they serve as an additional mechanism that *excludes conflicts* among accepted arguments, while in Example 4 (and sometimes also in Example 1) they actually *adapt for conflicts* which are inherent to the state of affairs. Clearly, such opposing situations require two different treatments, and in this paper we refer to both of them, namely, we consider coherence-based (or *conflict-free*) constrained systems on one hand and paraconsistent (or *conflict-tolerant*) systems on the other hand. In both cases we show how argumentation frameworks and integrity constraints are incorporated, define appropriate semantics for maintaining conflicts, and describe corresponding methods of representing and computing their consequences.

The rest of this paper is organized as follows: in the next section we review the basics behind Dung's abstract argumentation theory and recall the primary methods of giving it conflict-free and conflict-tolerant semantics. In Section 3 we consider constrained argumentation frameworks (CAFs). Again, we distinguish between cases in which conflicts should be dismissed and those in which conflicts may be accepted. In both cases we define appropriate semantics, compare them, are examine their basic properties. In Section 4 we consider some representation and computation aspects of reasoning with CAFs, and in Section 5 we conclude.

2. Preliminaries

2.1. Abstract Argumentation Frameworks and Their Semantics

Let us first recall the basics behind Dung's theory of abstract argumentation [1].

Definition 5. An *argumentation framework* is a pair $\mathcal{AF} = \langle \mathit{Args}, \mathit{Attack} \rangle$, where Args is a set (of arguments) and Attack is a relation on $\mathit{Args} \times \mathit{Args}$.

In what follows we shall assume that the argumentation frameworks are finite, that is, their sets of arguments are finite. When $(A, B) \in \mathit{Attack}$ we say that A *attacks* B (or that B is *attacked by* A). The set of arguments that are attacked by A is denoted by A^+ and the set of arguments that attack A is denoted by A^-. This may be extended to sets of arguments as follows: $\mathcal{E}^+ = \bigcup_{A \in \mathcal{E}} A^+$ is the set of arguments that are attacked by some argument in $\mathcal{E}$ and $\mathcal{E}^- = \bigcup_{A \in \mathcal{E}} A^-$ is the set of arguments that attack some argument in $\mathcal{E}$. We denote by $\mathrm{Def}(\mathcal{E})$ the set of arguments that are *defended by* $\mathcal{E}$, in the sense that each attacker of an argument in this set is counter-attacked by (an argument in) $\mathcal{E}$. Formally: $\mathrm{Def}(\mathcal{E}) = \{A \in \mathit{Args} \mid A^- \subseteq \mathcal{E}^+\}$.

The primary principles for accepting arguments in Dung-style argumentation are the following:

Definition 6. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework and let $\mathcal{E} \subseteq Args$ be a set of arguments.

- $\mathcal{E}$ is *conflict-free* (with respect to $\mathcal{AF}$) iff $\mathcal{E} \cap \mathcal{E}^+ = \emptyset$.
- $\mathcal{E}$ is an *admissible extension* (of $\mathcal{AF}$) iff it is conflict free and $\mathcal{E} \subseteq \mathrm{Def}(\mathcal{E})$.
- $\mathcal{E}$ is a *complete extension* (of $\mathcal{AF}$) iff it is conflict free and $\mathcal{E} = \mathrm{Def}(\mathcal{E})$.

Thus, conflict-freeness assures that no argument in the set is attacked by another argument in the set. Admissibility guarantees, in addition, that the set is self-defendant, and complete sets are admissible ones that defend exactly themselves. These principles are a cornerstone of a variety of extension-based semantics for an argumentation framework $\mathcal{AF}$, i.e., formalizations of sets of arguments that can collectively be accepted according to $\mathcal{AF}$. Some of these semantics are listed next (see also [1, 13, 14]).

Definition 7. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework and let $\mathcal{E} \subseteq Args$. Below, the minimum and maximum are taken with respect to set inclusion.

- $\mathcal{E}$ is a *grounded extension* of $\mathcal{AF}$ iff it is a minimal complete extension of $\mathcal{AF}$.
- $\mathcal{E}$ is a *preferred extension* of $\mathcal{AF}$ iff it is a maximal complete extension of $\mathcal{AF}$.
- $\mathcal{E}$ is a *stable extension* of $\mathcal{AF}$ iff it is a complete extension of $\mathcal{A}$ and $\mathcal{E}^+ = Args \setminus \mathcal{E}$.
- $\mathcal{E}$ is a *semi-stable extension* of $\mathcal{AF}$ iff it is a complete extension of $\mathcal{AF}$ where $\mathcal{E} \cup \mathcal{E}^+$ is maximal among all the complete extensions of $\mathcal{AF}$.

Argument acceptability may now be defined as follows:

Definition 8. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework, and let Sem be one type of the extensions (semantics) for $\mathcal{AF}$ considered in Definition 7 (that is, grounded, preferred, stable or semi-stable semantics). An argument $A \in Args$ is *credulously accepted* by Sem if it belongs to *some* Sem-extension of $\mathcal{AF}$; A is *skeptically accepted* by Sem if it belongs to *all* the Sem-extensions of $\mathcal{AF}$.

Example 9. Consider the framework $\mathcal{AF}_1$ of Figure 1. This framework has four admissible extensions: $\emptyset$, $\{A\}$, $\{B\}$ and $\{A, C\}$, three of them are complete: $\emptyset$, $\{B\}$ and $\{A, C\}$. It follows that $\emptyset$ is the grounded extension of $\mathcal{AF}_1$ and both of $\{B\}$ and $\{A, C\}$ are the preferred, stable and semi-stable extensions of $\mathcal{AF}_1$. In this case, then, according to the complete, preferred, stable and semi-stable semantics, none of the arguments is skeptically accepted, while A, B and C are credulously accepted.

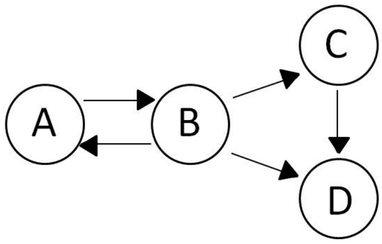

Figure 1: The argumentation framework $\mathcal{AF}_1$

Skeptical and credulous acceptance may be defined also with respect to other types of extensions. We refer, e.g., to [13, 14] for further details.

An alternative way to describe argumentation semantics is based on the concept of an *argument labeling*, defined next (see [15, 16]).

Definition 10. Let $\mathcal{AF} = \langle Args, Attack \rangle$ 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 $\mathcal{AF} = \langle Args, Attack \rangle$ can be seen as a partition of $Args$, following [15] we shall sometimes write it as a triple $\langle \mathsf{In}(lab), \mathsf{Out}(lab), \mathsf{Undec}(lab) \rangle$.

In a somewhat more logic-based fashion, labelings may be viewed as *valuations*. In what follows we denote by $\mathcal{L}_{Args}$ a propositional language whose atomic formulas are associated with the arguments of an argumentation framework $\langle Args, Attack \rangle$. A labeling in this context is then a truth-valued assignment for the atoms of $\mathcal{L}_{Args}$. We shall associate the label in with the truth value t that represents truth, the label out will be associated with the truth value f that represents falsity, and undec is associated with the middle (neutral) element $\perp$. Given a labeling lab on $\{\mathsf{in}, \mathsf{out}, \mathsf{undec}\}$ we shall denote by $\mathcal{LV}(lab)$ the associated valuation on $\{t, f, \perp\}$ and conversely: for a valuation ν on $\{t, f, \perp\}$ we denote by $\mathcal{VL}(\nu)$ the associated labeling on $\{\mathsf{in}, \mathsf{out}, \mathsf{undec}\}$.[1]

The following postulates allow to associate labelings also with extensions.

[1]The functions' names abbreviate their roles: $\mathcal{LV}$ stands for 'labelings to valuations' and $\mathcal{VL}$ stands for 'valuations to labelings'. We use similar notations for the other mappings defined in the sequel (see, e.g, Proposition 13 and Definition 19 below).

Definition 11. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework, lab an argument labeling for $Args$, and $A \in Args$. We consider the following conditions on lab:

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 all $B \in A^-$ it holds that $lab(B) = \mathsf{out}$ and there is no $B \in A^-$ such that $lab(B) = \mathsf{in}$.

Given a labeling lab of an argumentation framework $\langle Args, Attack \rangle$, we say that

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

Based on the concepts of conflict-free labelings and complete labelings, one may define labelings that correspond to the extensions considered in Definition 7.

Definition 12. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework and let lab be a complete labeling of $\mathcal{AF}$. Below, the minimum and the maximum are taken with respect to set inclusion.

- lab is a *grounded labeling* of $\mathcal{AF}$ iff $\mathsf{In}(lab)$ is minimal in $\{\mathsf{In}(l) \mid l$ is a complete labeling of $\mathcal{AF}\}$.
- lab is a *preferred labeling* of $\mathcal{AF}$ iff $\mathsf{In}(lab)$ is maximal in $\{\mathsf{In}(l) \mid l$ is a complete labeling of $\mathcal{AF}\}$.
- lab is a *stable labeling* of $\mathcal{AF}$ iff $\mathsf{Undec}(lab) = \emptyset$.
- lab is a *semi-stable labeling* of $\mathcal{AF}$ iff $\mathsf{Undec}(lab)$ is minimal in $\{\mathsf{Undec}(l) \mid l$ is a complete labeling of $\mathcal{AF}\}$.

The following correspondence between extensions and labelings is shown in [16]:

Proposition 13. *Let $\mathcal{AF} = \langle Args, Attack \rangle$ 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 $\mathcal{LE} : \mathcal{CFL} \to \mathcal{CFE}$, defined by $\mathcal{LE}(lab) = \mathsf{In}(lab)$, and the function $\mathcal{EL}_{\mathcal{AF}} : \mathcal{CFE} \to \mathcal{CFL}$, defined by $\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}) = \langle \mathcal{E}, \mathcal{E}^+, Args \setminus (\mathcal{E} \cup \mathcal{E}^+) \rangle$. It holds that:*

1. *If $\mathcal{E}$ is an admissible (respectively, complete) extension, then $\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is an admissible (respectively, complete) labeling.*

2. *If lab is an admissible (respectively, complete) labeling, then $\mathcal{LE}(lab)$ is an admissible (respectively, complete) extension.*
3. *When the domain and range of $\mathcal{EL}_{\mathcal{AF}}$ and $\mathcal{LE}$ 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.*

Similar correspondence hold between the extensions in Definition 7 and the corresponding labelings in Definition 12 (see [16]).

2.2. Conflict-Tolerant Semantics

As we noted in the introduction, for properly reflecting real-life situations it is occasionally required to abandon the conflict-freeness assumption behind standard argumentation semantics, so it might happen that accepted arguments attack each other. When constraints are incorporated, conflict tolerance is sometimes essential, since – as we shall see shortly – even constraints of a very simple form may imply mutual attacks among accepted arguments. To handle this we incorporate the conflicting-tolerant semantics for argumentation frameworks introduced in [17, 18]. In this section we briefly recall this semantics.[2]

The most straightforward way of maintaining conflicts while still being as faithful as possible to the conflict-free semantics considered previously is by lifting the conflict-freeness requirement in Definition 6, while keeping the other properties in the same definition. Thus, any argument in an extension must still be defended (to avoid arbitrary acceptance of arguments).

Definition 14. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework and let $\mathcal{E} \subseteq Args$.

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

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. Note that, as in the case of conflict-free semantics, p-grounded and p-preferred extensions may be defined by taking, respectively, the subset-minimal and the subset-maximal p-complete extensions.

Example 15. Consider again the framework $\mathcal{AF}_1$ of Example 9.

1. The p-admissible extensions of $\mathcal{AF}_1$ are $\emptyset$, $\{A\}$, $\{B\}$, $\{A, B\}$, $\{A, C\}$, $\{A, B, C\}$, $\{A, B, D\}$ and $\{A, B, C, D\}$.

[2]For the proofs of the propositions in this sections see [18].

2. The p-complete extensions of $\mathcal{AF}_1$ are $\emptyset$, $\{B\}$, $\{A, C\}$ and $\{A, B, C, D\}$.

Example 16. 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 in this case), and $\{A, B, C\}$.

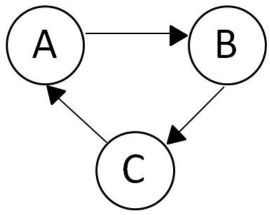

Figure 2: The argumentation framework $\mathcal{AF}_2$

Proposition 17. *There exists a nonempty p-complete extension (and so there is a nonempty p-admissible extension) for every argumentation framework.*

As in the case of conflict-free semantics, there is a dual way of representing p-admissible and p-complete extensions, which is based on labeling functions. This time, however, *four* labels are necessary for depicting the possible states of an argument.

Definition 18. Let $\mathcal{AF} = \langle \mathit{Args}, \mathit{Attack} \rangle$ be an argumentation framework. A *four-states labeling* for $\mathcal{AF}$ is a complete function $lab : \mathit{Args} \to \{\mathsf{in}, \mathsf{out}, \mathsf{none}, \mathsf{both}\}$. Again, we shall write $\mathsf{In}(lab)$ for $\{A \in \mathit{Args} \mid lab(A) = \mathsf{in}\}$ and $\mathsf{Out}(lab)$ for $\{A \in \mathit{Args} \mid lab(A) = \mathsf{out}\}$. Also, $\mathsf{None}(lab)$ is the set $\{A \in \mathit{Args} \mid lab(A) = \mathsf{none}\}$ and $\mathsf{Both}(lab)$ is the set $\{A \in \mathit{Args} \mid lab(A) = \mathsf{both}\}$.

As before, a labeling function reflects the state of mind of the reasoner regarding each argument in $\mathcal{AF}$: $\mathsf{In}(lab)$ is the set of arguments that one accepts, $\mathsf{Out}(lab)$ is the set of arguments that one rejects, $\mathsf{None}(lab)$ is the set of arguments that may neither be accepted nor rejected, and $\mathsf{Both}(lab)$ is the set of arguments that have both supportive and rejective evidences. In the sequel we shall sometimes represent a 4-states labeling lab by the quadruple $\langle \mathsf{In}(lab), \mathsf{Out}(lab), \mathsf{None}(lab), \mathsf{Both}(lab) \rangle$.

Given a labeling lab on $\{\mathsf{in}, \mathsf{out}, \mathsf{both}, \mathsf{none}\}$ we shall denote by $p\mathcal{LV}(lab)$ the associated valuation on $\{t, f, \top, \bot\}$ and conversely: for a valuation ν on $\{t, f, \top, \bot\}$ we denote by $p\mathcal{VL}(\nu)$ the associated labeling on $\{\mathsf{in}, \mathsf{out}, \mathsf{both}, \mathsf{none}\}$. Here, $\top$ is the truth value that intuitively represents contradictory information and $\bot$ is the truth value that intuitively represents lack of information (see Section 3.2).

Again, one may switch between extensions and labelings using appropriate mapping functions:

Definition 19. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework.

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

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

  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

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

As in the conflict-free case, special labeling postulates are defined for guaranteeing a one-to-one correspondence between extension-based and labeling-based conflict-tolerant semantics.

Definition 20. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework.

- A p-labeling *lab* for $\mathcal{AF}$ is *p-admissible* if it satisfies the following rules:

  pIn If $lab(A) = \mathsf{in}$ then $lab(B) = \mathsf{out}$ for all $B \in A^-$.

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

  pBoth If $lab(A) = \mathsf{both}$ then $lab(B) \in \{\mathsf{out}, \mathsf{both}\}$ for all $B \in A^-$ and $lab(B) = \mathsf{both}$ for some $B \in A^-$.

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

- A p-labeling *lab* for $\mathcal{AF}$ is *p-complete* if it satisfies the following rules:

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

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

  pBoth$^+$ $lab(A) = \mathsf{both}$ iff $lab(B) \in \{\mathsf{out}, \mathsf{both}\}$ for all $B \in A^-$ and $lab(B) = \mathsf{both}$. for some $B \in A^-$.

  pNone$^+$ $lab(A) = \mathsf{none}$ iff $lab(B) \in \{\mathsf{out}, \mathsf{none}\}$ for all $B \in A^-$ and $lab(B) = \mathsf{none}$ for some $B \in A^-$.

Proposition 21. *Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework.*

- *If $\mathcal{E}$ is a p-admissible extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-admissible labeling of $\mathcal{AF}$ and if lab is a p-admissible labeling of $\mathcal{AF}$ then $p\mathcal{LE}(lab)$ is a p-admissible extension for $\mathcal{AF}$. Moreover, in this case $p\mathcal{LE}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})) = \mathcal{E}$ and $p\mathcal{EL}_{\mathcal{AF}}(p\mathcal{LE}(lab)) = lab$. Thus, the functions $p\mathcal{EL}_{\mathcal{AF}}$ and $p\mathcal{LE}$, restricted to the p-admissible labelings and the p-admissible extensions of $\mathcal{AF}$, are bijective, and are each other's inverse.*

- *If $\mathcal{E}$ is a p-complete extension of $\mathcal{AF}$ then $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-complete labeling of $\mathcal{AF}$ and if lab is a p-complete labeling of $\mathcal{AF}$ then $p\mathcal{LE}(lab)$ is a p-complete extension for $\mathcal{AF}$. Moreover, in this case $p\mathcal{LE}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})) = \mathcal{E}$ and $p\mathcal{EL}_{\mathcal{AF}}(p\mathcal{LE}(lab)) = lab$. Thus, the functions $p\mathcal{EL}_{\mathcal{AF}}$ and $p\mathcal{LE}$, restricted to the p-complete labelings and the p-complete extensions of $\mathcal{AF}$, are bijective, and are each other's inverse.*

Example 22. The eight p-admissible labelings of $\mathcal{AF}_1$ (Figure 1) are listed in Table 1. These labelings correspond to the eight p-admissible extensions of $\mathcal{AF}_1$, listed in Example 15. Four of these labelings are also p-complete (see the rightmost column in the table). Again, these labelings correspond to the four p-complete extensions of $\mathcal{AF}_1$ listed in Example 15, as indeed suggested by Proposition 21.

p-*lab*	A	B	C	D	induced p-extension	p-complete?
1	none	none	none	none	$\{\}$	$+$
2	in	out	none	none	$\{A\}$	$-$
3	out	in	out	out	$\{B\}$	$+$
4	in	out	in	out	$\{A, C\}$	$+$
5	both	both	out	out	$\{A, B\}$	$-$
6	both	both	both	out	$\{A, B, C\}$	$-$
7	both	both	out	both	$\{A, B, D\}$	$-$
8	both	both	both	both	$\{A, B, C, D\}$	$+$

Table 1: The p-admissible labelings of $\mathcal{AF}_1$

3. Constrained Argumentation Frameworks

We now consider constrained argumentation frameworks. These are argumentation frameworks augmented with set of formulas (the 'constraints') that should be satisfied by any extension or labeling of the framework. As indicated in the introduction, such formulas are useful for introducing additional knowledge that cannot be extracted from the framework itself, such as arguments dependencies, relations among arguments that are not depicted by the attack relation, preferences among arguments, and so forth. Below, we distinguish between two cases: the first one, considered in Section 3.1, is based on 3-valued, conflict-free semantics. The other case, considered in Section 3.2, relies on 4-valued, conflict-tolerant semantics. The choice which approach to use depends, of-course, on the situation at hand and on the plausibility of accommodating contradictory data and conflicting arguments.

3.1. Three-Valued Conflict-Free Semantics

First, we consider constrained argumentation frameworks whose semantics is conflict-free. The constraints in such frameworks are expressed by formulas in the language $\mathcal{L}_{Args}$, whose atomic formulas are associated with the arguments *Args* of the framework. In addition, $\mathcal{L}_{Args}$ contains the connectives $\vee$, $\wedge$, $\supset$, $\neg$, and the propositional constants t, f, and u that intuitively correspond to the three states in, out, undec, of conflict-free labeling functions. As noted in Section 2.1, a conflict-free labeling *lab* for an argumentation framework $\mathcal{AF} = \langle Args, Attack \rangle$ corresponds to a truth assignment (valuation) $\mathcal{LV}(lab)$ of values from the atoms of $\mathcal{L}_{Args}$ to $\{t, f, \bot\}$.[3] These valuations may be extended to complex formulas in $\{\vee, \wedge, \supset, \neg\}$ by Kleene's three-valued interpretations for the disjunction $\vee$, conjunction $\wedge$ and negation $\neg$ (see [19]), and by Słupecki's interpretation for the implication $\supset$ (see [20, 21]),[4] as follows:

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

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

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

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

As usual in this context, we say that a 3-valued valuation ν *satisfies* (or, is a 3-valued *model* of) a set of formulas $\mathcal{S}$, if $\nu(\psi) = t$ for every $\psi \in \mathcal{S}$. We denote the set of the 3-valued models of $\mathcal{S}$ by $mod^3(\mathcal{S})$.

Example 23. Consider again Example 3, and denote by A_1, A_2 and A_3 the three arguments mentioned there. The restriction that these arguments cannot be accepted together may be enforced by adding, e.g., the integrity constraint $(A_1 \wedge A_2 \wedge A_3) \supset \mathsf{f}$. Indeed, ν is a 3-valued model of this formula iff $\nu(A_1 \wedge A_2 \wedge A_3) \neq t$.[5]

A natural requirement from constraints applied to argumentation frameworks with conflict-free semantics is that they would have admissible interpretations, namely: the constraints themselves should not be contradictory and every argument that is satisfied by their models shouldn't be exposed to undefended attacks (indeed, these are primary requirements from any accepted

[3] We assume that the propositional constants t, f, and u of $\mathcal{L}_{Args}$ are assigned the truth value t and f and $\bot$ (respectively) by every valuation.

[4] As indicated, e.g., in [22], the material implication defined by $A \to B = \neg A \vee B$, is not a proper choice for an implication connective in this case, since e.g. $\nu(A \to A) \neq t$ for every valuation ν for which $\nu(A) = \bot$.

[5] Note that in the 3-valued case the formula $\neg(A_1 \wedge A_2 \wedge A_3)$ is too restrictive for our purpose, since it requires that $A_1 \wedge A_2 \wedge A_3$ must be falsified (i.e., its value should be f), while we want to require that $A_1 \wedge A_2 \wedge A_3$ is not satisfied (i.e., its value should be either f or $\bot$).

arguments, and so they obviously apply to those that *must* be accepted). This leads to the next definitions.

Definition 24. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework. A set of formulas *Const* in $\mathcal{L}_{Args}$ is called *admissible* (for $\mathcal{AF}$), if it has a 3-valued model ν so that $\mathcal{VL}(\nu)$ is an admissible labeling of $\mathcal{AF}$.

Definition 25. A *constrained argumentation framework* (CAF, for short) is a triple $\mathcal{CAF} = \langle Args, Attack, Const \rangle$, where $\mathcal{AF} = \langle Args, Attack \rangle$ is an argumentation framework and *Const* (the constraints) is a set of formulas in $\mathcal{L}_{Args}$ which is admissible for $\mathcal{AF}$.

Definition 26. Let $\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a constrained argumentation framework and let Sem be a conflict-free semantics for $\mathcal{AF} = \langle Args, Attack \rangle$.

1. We say that *lab* is a Sem-*labeling* of $\mathcal{CAF}$ if it is a Sem-labeling of $\mathcal{AF}$ and $\mathcal{LV}(lab)$ is a 3-valued model of *Const*.
2. We say that $\mathcal{E}$ is a Sem-*extension* of $\mathcal{CAF}$ if $\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a Sem-labeling of $\mathcal{CAF}$.

By Proposition 13, we have:

Proposition 27. *Let $\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a constrained argumentation framework. For every conflict-free semantics Sem for $\mathcal{AF} = \langle Args, Attack \rangle$ we have that $\mathcal{E} \subseteq Args$ is a Sem-extension of $\mathcal{CAF}$ iff it is a Sem-extension of $\mathcal{AF}$ and $\mathcal{LV}(\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ satisfies Const.*

Example 28. Consider the constrained argumentation framework $\mathcal{CAF}_1^{A \vee B}$ that consists of the argumentation framework $\mathcal{AF}_1$ of Figure 1 and the constraint $A \vee B$. For every semantics Sem considered in Section 2.1 we have that the Sem-extensions of $\mathcal{CAF}_1^{A \vee B}$ coincide with the *nonempty* Sem-extensions of $\mathcal{AF}_1$ (see also Example 9). In particular, the complete extensions of $\mathcal{CAF}_1^{A \vee B}$ are $\{A, C\}$ and $\{B\}$ (since $\langle \{A, C\}, \{B, D\}, \{\} \rangle$ and $\langle \{B\}, \{A, C, D\}, \{\} \rangle$ are the only complete labelings of $\mathcal{AF}_1$ so that the 3-valued truth assignments on $\{A, B, C, D\}$ that are associated with them satisfy $A \vee B$).

Note 29. The constrained argumentation frameworks considered in [18] and in [23] are a particular case of those in Definition 25, where *Const* is restricted to atomic formulas only.

Constrained argumentation frameworks are also considered by Coste-Marquis, Devred and Marquis in [24]. The main difference is that in [24] the interpretations are determined by completeness semantics: a subset $\mathcal{E} \subseteq Args$ is associated with a two-valued valuation that is induced by its completion $\hat{\mathcal{E}} = \{A \mid A \in \mathcal{E}\} \cup \{\neg A \mid A \notin \mathcal{E}\}$, and satisfiability of constraints is with respect to two-valued semantics. It follows, e.g., that a constraint of the form $A \vee \neg A$ is useless according to [24] (since it is always satisfied), while in our 3-valued semantics this constraint indicates that the argument A cannot have a neutral status. What is more, the use of 3-valued semantics allows us to distinguish between different restrictions on arguments: the constraint $\neg A$ means that A should be rejected, while the constraint $A \supset \mathsf{f}$ is a somewhat weaker demand, that A should not be accepted, and so its status may be undecided.[6] We thus believe that a 3-valued semantics for the constraints is more in line with standard 3-state semantics of argumentation frameworks.

Another difference between the approaches is that in our case the integrity constraints are admissible. This assures Propositions 33 and 37 below, which do not hold in the case of [24], where non-empty extensions for CAFs may not exist. Recently, Booth et al. [11] provided a method for generating non-empty conflict-free extensions for constrained argumentation frameworks, but the price for that is a waiving of the principle of admissibility, so in their formalism not only the integrity constraints, but also the extensions themselves may not be admissible. Thus, for instance, the addition of the (non-admissible) constraint $A \vee B \vee C$ to the argumentation framework $\mathcal{AF}_2$ of Figure 2 would yield, according to [11], three extensions $\{A\}$, $\{B\}$, $\{C\}$, each one is conflict-free, but neither of them is admissible.

Obviously, when the constraints are weakened (respectively, strengthened), the set of extensions may be expanded (respectively, reduced):

Proposition 30. *Suppose that $\mathcal{CAF}_1 = \langle Args, Attack, Const_1\rangle$ and $\mathcal{CAF}_2 = \langle Args, Attack, Const_2\rangle$ are two CAFs such that $mod^3(Const_1) \subseteq mod^3(Const_2)$ and let* Sem *be one of their conflict-free semantics discussed previously. Then every* Sem*-labeling/extension of $\mathcal{CAF}_1$ is also a* Sem*-labeling/extension of $\mathcal{CAF}_2$.*

Proof. Let *lab* be a Sem-labeling of $\mathcal{CAF}_1$. In particular, *lab* is a Sem-labeling of $\langle Args, Attack\rangle$, and so $\mathcal{LV}(lab)$ is a model of $Const_1$. Since $mod^3(Const_1) \subseteq mod^3(Const_2)$, $\mathcal{LV}(lab)$ is also a model of $Const_2$, thus *lab* is a Sem-labeling of $\mathcal{CAF}_2$ as well. The considerations regarding Sem-extensions are similar. $\square$

The next proposition shows that the relations among basic conflict-free semantics for argumentation frameworks carry on to CAFs.

Proposition 31. *Let $\mathcal{CAF} = \langle Args, Attack, Const\rangle$ be a constrained argumentation framework. Then: (a) if the grounded extension of $\mathcal{CAF}$ exists, it is contained in every complete extension of $\mathcal{CAF}$, (b) every stable extension of $\mathcal{CAF}$ is a semi-stable extension of $\mathcal{CAF}$, (c) every semi-stable extension of $\mathcal{CAF}$ is a preferred extension of $\mathcal{CAF}$, and (d) every preferred extension of $\mathcal{CAF}$ is a complete extension of $\mathcal{CAF}$. Similar relations hold for the corresponding labeling functions.*

Proof. All the items follow from Proposition 27 and the similar relations that hold among the relevant extensions of $\mathcal{AF} = \langle Args, Attack\rangle$. By the one-to-one correspondence between complete [respectively: grounded, preferred, stable,

[6] Indeed, a model of the first constraint must assign f to A, while in the second case A may be assigned any value other than t (i.e., either f or $\perp$).

semi-stable] extensions and complete [respectively: grounded, preferred, stable, semi-stable] labelings, these results hold also for the corresponding labelings. □

Since every Sem-extension of a constrained argumentation framework is in particular a Sem-extension of the corresponding argumentation framework, we immediately have the following corollary of the last proposition.

Corollary 32. *Let $\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a constrained argumentation framework for the argumentation framework $\mathcal{AF} = \langle Args, Attack \rangle$. Then: (a) if the grounded extension of $\mathcal{CAF}$ exists, it is contained in every complete extension of $\mathcal{AF}$, (b) every stable extension of $\mathcal{CAF}$ is a semi-stable extension of $\mathcal{AF}$, (c) every semi-stable extension of $\mathcal{CAF}$ is a preferred extension of $\mathcal{AF}$, and (d) every preferred extension of $\mathcal{CAF}$ is a complete extension of $\mathcal{AF}$. Similar relations hold for the corresponding labeling functions.*

We now turn to the issue of the *existence* of extensions for CAFs.

Proposition 33. *Every CAF has an admissible extension/labeling.*

Proof. Immediate from the fact that *Const* is admissible. □

Next, we show that complete extensions and labelings are guaranteed for CAFs whose constraints are in the language of $\{\vee, \wedge, \neg\}$ (see Proposition 37). For this, we first need a definition and a lemma.

Definition 34. We denote by $\leq_k$ the partial order on $\{t, f, \bot\}$ in which t and f are the (incomparable) $\leq_k$-maximal elements and $\bot$ is the $\leq_k$-minimal element. Accordingly, we define a partial order (with the same notation) on 3-valued valuations by pointwise comparisons of their atomic assignments: given 3-valued valuations ν and μ on *Args*, we denote by $\mu \geq_k \nu$ that $\mu(A) \geq_k \nu(A)$ for every $A \in Args$.

Lemma 35. *Let Const be a set of formulas in the language of $\{\vee, \wedge, \neg\}$. If ν is a 3-valued model of Const and $\mu \geq_k \nu$, then μ is a 3-valued model of Const as well.*

Proof. By induction on the structure of a formula ψ in the language of $\{\vee, \wedge, \neg\}$ one can show that if $\mu \geq_k \nu$ (i.e., if $\mu(A) \geq_k \nu(A)$ for every $A \in Args$), then $\mu(\psi) \geq_k \nu(\psi)$ as well. This is true, in particular, for every constraint ψ. Thus, if $\nu(\psi) = t$ for $\psi \in Const$, also $\mu(\psi) = t$. □

Note 36. When $\supset$ is a connective in the language, the lemma does not hold any longer. For instance, a valuation ν such that $\nu(p) = \bot$ is a model of $p \supset \neg p$, but a valuation μ for which $\mu(p) = t$ is not a model of this formula, although $\mu(p) >_k \nu(p)$.

Proposition 37. *Every CAF whose constraints are in the language of $\{\vee, \wedge, \neg\}$ has a complete extension/labeling.*

Proof. Let $\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a constrained argumentation framework. By Proposition 33 there is an admissible labeling lab for $\mathcal{CAF}$. In particular, lab is an admissible labeling of $\mathcal{AF} = \langle Attack, Const \rangle$ and $\mathcal{LV}(lab)$ is a 3-valued model of $Const$. If lab is also a complete labeling of $\mathcal{AF}$, it is a complete labeling of $\mathcal{CAF}$ and so we are done. Otherwise, it is well-known that lab can be 'completed', that is, turned into a complete labeling lab_c of $\mathcal{AF}$, by changing some of its undec-assignments to in or out-assignments (so that the rule **Neither** in Definition 11 will be satisfied without violating the rules **Pos2** and **Neg** in the same definition). In particular, for every $A \in Args$ it holds that if $lab(A) = \mathsf{in}$ so $lab_c(A) = \mathsf{in}$ and if $lab(A) = \mathsf{out}$ so $lab_c(A) = \mathsf{out}$. It follows that for every $A \in Args$, $\mathcal{LV}(lab) = t$ implies that $\mathcal{LV}(lab_c) = t$ and $\mathcal{LV}(lab) = f$ implies that $\mathcal{LV}(lab_c) = f$. Thus, $\mathcal{LV}(lab) \leq_k \mathcal{LV}(lab_c)$. Since $\mathcal{LV}(lab)$ is a 3-valued model of $Const$ we have, by Lemma 35, that $\mathcal{LV}(lab_c)$ satisfies $Const$ as well, and so lab_c is a complete labeling of $\mathcal{CAF}$. $\square$

The next example shows that the condition in Proposition 37 is indeed necessary.

Example 38. Consider the constrained argumentation framework that consists of the argumentation framework in Figure 3 and the constraint $A \wedge (C \supset \mathsf{f})$.

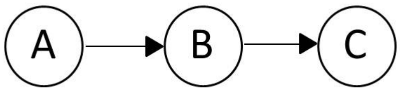

Figure 3: The argumentation framework for Example 38

We have that $\langle \{A\}, \{B\}, \{C\} \rangle$ is an admissible labeling for this constrained framework (since the valuation ν that is associated with this labeling, in which $\nu(A) = t$, $\nu(B) = f$ and $\nu(C) = \perp$, satisfies the constraint), however, there is no complete labeling of this framework for which the constraint holds.

Note 39. Let $\models^3$ be the standard 3-valued satisfiability entailment, defined by $\Gamma \models^3 \Delta$ if $mod^3(\Gamma) \subseteq mod^3(\Delta)$. Then, in fact, Proposition 37 is no longer true not only for $\supset$ (as Example 38 shows), but also for *every* 3-valued implication $\mapsto$ which is $\models^3$-deductive and is a conservative extension to the 3-valued case of the material implication. Indeed, since $\mapsto$ is $\models^3$-deductive, $\perp \mapsto f = t$ (Otherwise $\perp \mapsto f \neq t$, and so, while $A \wedge \neg A \models^3 \mathsf{f}$, we have that $\not\models^3 (A \wedge \neg A) \mapsto \mathsf{f}$ because $\nu(A) = \perp$ is a counter-model). But if $\perp \mapsto f = t$ then again the last example shows that Proposition 37 fails for languages with $\mapsto$. Indeed, $\nu(A) = t$, $\nu(B) = f$ and $\nu(C) = \perp$ would still satisfy the constraint in that example, but for no complete extension of the argumentation framework of Figure 3 the constraint holds, otherwise both A and C would have been labeled in, so the associated valuation would assign t to both of them, and to satisfy $C \mapsto \mathsf{f}$ we would need to have $t \mapsto f = t$ (which is impossible for any conservative extension of the material implication).

We turn now to grounded extensions. This time, as the next example shows, their existence is not guaranteed even for CAFs whose constraints are in the language without $\supset$. The example also shows that (in contrast to standard argumentation frameworks) a subset-minimal complete extension of a CAF need not be its grounded extension.[7]

Example 40. The constraint argumentation framework $\mathcal{CAF}_1^{A\vee B}$ of Example 28 does not have a grounded extension since the grounded extension of $\mathcal{AF}_1$ is the emptyset, but the associated 3-valued valuation, which is the uniform $\bot$-assignment, does not satisfy the constraint $A\vee B$. This is also the reason that the argumentation framework $\mathcal{CAF}_4^{A\vee B}$, consisting of the argumentation framework $\mathcal{AF}_4$ in Figure 4 and the same integrity constraint, does not have a grounded extension. We note that both of $\langle\{A,C\},\{B,D\},\{\}\rangle$ and $\langle\{B,D\},\{A,C\},\{\}\rangle$ are complete extensions of $\mathcal{AF}_4$, and the 3-valued valuations that are associated with them satisfy $A\vee B$. Therefore, $\mathcal{CAF}_4^{A\vee B}$ has two complete extensions, both of which are minimal (with respect to the subset relation) among the complete extensions of $\mathcal{AF}_4$ that satisfy the constraint, but neither of them is a grounded extension of $\mathcal{AF}_4$.

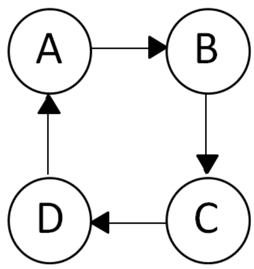

Figure 4: The argumentation framework for Example 40

Clearly, when a grounded extension of a CAF does exist, it is the unique subset-minimal complete extension of that CAF.

3.2. Four-Valued Conflict-Tolerant Semantics

We now turn to constrained argumentation frameworks whose semantics is conflict-tolerant. Again, the constraints are expressed by a language whose atomic formulas are associated with the arguments *Args* of the framework and whose connectives are in $\vee,\wedge,\supset,\neg$. In addition, the language contains the propositional constants t, f, n, and b that intuitively correspond to the four

[7] Recall that by Proposition 27, $\mathcal{E}$ is the grounded extension of $\mathcal{CAF}=\langle\mathcal{AF},\mathit{Const}\rangle$ if it is the grounded extension of $\mathcal{AF}$ and $\mathcal{LV}(\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ satisfies *Const*.

states $\{\mathsf{in}, \mathsf{out}, \mathsf{none}, \mathsf{both}\}$ of conflict-tolerant labeling functions. In what follows we shall continue to denote such languages by $\mathcal{L}_{Args}$. Again, as noted in Section 2.1, a conflict-tolerant labeling lab for an argumentation framework $\mathcal{AF} = \langle Args, Attack \rangle$ corresponds to a truth assignment (valuation) $p\mathcal{LV}(lab)$ of values in $\{t, f, \bot, \top\}$ to the atoms of $\mathcal{L}_{Args}$. These valuations are extended to complex formulas with connectives in $\{\vee, \wedge, \supset, \neg\}$ by the following Belnap's four-valued interpretations for the disjunction $\vee$, conjunction $\wedge$ and negation $\neg$ (see [25]), and by D'Ottaviano and da-Costa's interpretation for the implication $\supset$ (see [20, 26, 27]).[8]

$\vee$	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
$\top$	$\top$
$\bot$	$\bot$

Following [22, 25], we say that a valuation ν *satisfies* (or, is a 4-valued *model* of) a set of formulas $\mathcal{S}$ if $\nu(\psi) \in \{t, \top\}$ for every $\psi \in \mathcal{S}$. We denote the set of the 4-valued models of $\mathcal{S}$ by $mod^4(\mathcal{S})$.

Since conflict-tolerant semantics permits mutual attacks among accepted arguments, the integrity constraints in such cases may be contradictory. Accordingly, we relax the assumptions on plausible integrity constraints compared to those taken in the previous section, and now only require that they would have p-admissible interpretations (so their accepted arguments shouldn't be exposed to undefended attacks).

Definition 41. Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework. A set of formulas $Const$ in $\mathcal{L}_{Args}$ is called *p-admissible* (for $\mathcal{AF}$), if it has a 4-valued model ν so that $\mathcal{VL}(\nu)$ is a p-admissible labeling of $\mathcal{AF}$.

Example 42. The formula $\Psi_1 = A \wedge \neg C \wedge (C \supset \mathsf{f})$ is not p-admissible (and so it is not admissible) for the argumentation framework in Figure 3, because it requires that A will be accepted and C will be rejected at the same time. Indeed, any 4-valued model of this formula must assign f to C, thus the associated labeling assigns out to C. If this labeling were p-admissible, then by **pOut** B would have been assigned either in or both. In turn, this means that A should have been assigned either out or both, and so either **pOut** or **pBoth** were violated (respectively) when applied to A (since A does not have any attacker).

In contrast, the formula $\Psi_2 = A \wedge (C \supset \mathsf{f})$ *is* p-admissible for the argumentation framework in Figure 3, since it is already admissible for this framework (see Example 38). Intuitively, the difference between the two constraints is that unlike Ψ_1, which requires the rejection of C, Ψ_2 poses a weaker constraint on C, according to which C may as well be undecided.

[8] As indicated, e.g., in [22], these interpretations are natural generalizations of the dual 3-valued interpretations for the same connectives, considered in the previous section.

Definition 43. A *paraconsistent constrained argumentation framework* (pCAF, for short) is a triple $p\mathcal{CAF} = \langle Args, Attack, Const \rangle$, where $\mathcal{AF} = \langle Args, Attack \rangle$ is an argumentation framework and *Const* (the constraints) is a set of formulas in $\mathcal{L}_{Args}$ that is p-admissible for $\mathcal{AF}$.

Note 44. Since every admissible set for $\mathcal{AF}$ is also a p-admissible for $\mathcal{AF}$, we have that every CAF is also a pCAF (but not the other way around).

Definition 45. Let $p\,\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a p-constrained argumentation framework and let Sem be a conflict-tolerant semantics for $\mathcal{AF} = \langle Args, Attack \rangle$.

1. We say that lab is a Sem-*labeling* of $p\,\mathcal{CAF}$ if it is a Sem-labeling of $\mathcal{AF}$ and $p\mathcal{LV}(lab)$ is a 4-valued model of *Const*.
2. We say that $\mathcal{E}$ is a Sem-*extension* of $p\mathcal{CAF}$ if $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a Sem-labeling of $p\,\mathcal{CAF}$.

By Proposition 21 we have:

Proposition 46. *Let* $p\,\mathcal{CAF} = \langle Args, Attack, Const \rangle$ *be a p-constrained argumentation framework. For every conflict-tolerant semantics* Sem *for* $\mathcal{AF} = \langle Args, Attack \rangle$*, we have that* $\mathcal{E} \subseteq Args$ *is a* Sem*-extension of* $p\,\mathcal{CAF}$ *iff it is a* Sem*-extension of* $\mathcal{AF}$ *and* $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ *satisfies Const.*

Example 47. Consider again the argumentation framework $\mathcal{AF}_1$ of Figure 1 and the constraint $A \wedge B$. Since this constraint is not conflict-free for $\mathcal{AF}_1$ (i.e., no 3-valued valuation that satisfies it is associated with a conflict-free labeling of $\mathcal{AF}_1$), it is *not* admissible for $\mathcal{AF}_1$. However, this constraint *is* p-admissible for $\mathcal{AF}_1$ (since, e.g., the four-states labeling lab that assigns both to A and to B and out to C and to D is a p-admissible labeling of $\mathcal{AF}_1$, and $p\mathcal{LV}(lab)$ is a 4-valued model of $A \wedge B$). Therefore, $\mathcal{CAF}_1^{A\wedge B}$, obtained from $\mathcal{AF}_1$ and the constraint $A \wedge B$, is a pCAF. This pCAF has four p-admissible extensions: $\{A, B\}$, $\{A, B, C\}$, $\{A, B, D\}$ and $\{A, B, C, D\}$, the latter is also p-complete.

Example 48. Following Example 1, we consider the next set of rules (a variant of an example from [28]):

A "The bacteria in the blood is of type X and so a bacteria is present, but it cannot be of type Y"

B "The bacteria in the blood is of type Y and so a bacteria is present, but it cannot be of type X"

C "There is no bacterial infection thus no further medical examinations are required"

D "Further medical examinations are required and so another visit to the clinic should be scheduled"

Figure 5 shows an argumentation framework that depicts the interactions among

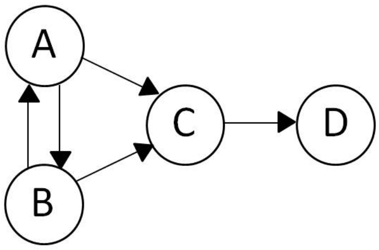

Figure 5: The argumentation framework for Example 48

these rules. Here, an argument attacks another if the consequence of the former contradicts an assumption of the latter.

Now, suppose that two blood tests of a patient indicate that a bacteria of a certain type is present, but each one indicates that the bacteria is of a different type: one indicates that it is of type X and the other one indicates that it is of type Y. Obviously, at least one of the tests is erroneous. Assuming that other blood tests are not available and that further tests cannot be taken, what can still be inferred in this case? One way of verifying this is to consider the pCAF that is obtained by the framework of Figure 5 and the constraint $A \wedge B$.[9] This pCAF has three p-admissible labelings, all of them assign the label both to A and to B (since both of these arguments should be accepted although they attack each other), but they differ regarding the statuses of the other arguments: one indicates that both C and D are contradictory, another one rejects C and accepts D, and the third one rejects C and labels D as undecided. Despite the inconsistency, then, the negation of argument C holds in all of the labelings, while neither D nor its negation are acceptable. This may be intuitively explained by the facts that a bacterial infection was detected (and so argument C is not relevant and should not be accepted), but according to the available information this does not necessarily mean that further medical examinations are required (thus D does not necessarily hold).

The next proposition is the dual, for conflict-tolerant semantics, of Proposition 30.

Proposition 49. *Suppose that* $p\mathcal{CAF}_1 = \langle Args, Attack, Const_1\rangle$ *and* $p\mathcal{CAF}_2 = \langle Args, Attack, Const_2\rangle$ *are two pCAFs such that* $mod^4(Const_1) \subseteq mod^4(Const_2)$ *and let* Sem *be one of their conflict-tolerant semantics discussed previously. Then every* Sem*-labeling/extension of* $p\,\mathcal{CAF}_1$ *is also a* Sem*-labeling/extension of* $p\,\mathcal{CAF}_2$*.*

[9]We need a pCAF here since $A \wedge B$ is not conflict-free for the argumentation framework of Figure 5. Similar considerations to those in Example 47 show that this argumentation framework together with the constraint $A \wedge B$ is indeed a pCAF.

Proof. Similar to that of Proposition 30. $\square$

We now turn to the issue of the *existence* of acceptable sets of arguments for pCAFs. It turns out that the situation is quite similar to that of CAFs (cf. Propositions 33 and 37).

Proposition 50. *Every pCAF has a p-admissible extension/labeling.*

Proof. Immediate from the fact that *Const* is p-admissible. $\square$

We now show that, like the 3-valued case, p-complete extensions and labelings are guaranteed for pCAFs whose constraints are in the language of $\{\vee, \wedge, \neg\}$ (Proposition 56). For this, we first need a definition and two lemmas.

Definition 51. We define partial orders on 4-valued valuations and 4-states labelings as follows:

- We denote by $\leq_k$ the partial order on $\{t, f, \top, \bot\}$ in which $\bot$ is the $\leq_k$-minimal element, $\top$ is the $\leq_k$-maximal element, and t and f are (incomparable) intermediate elements. Accordingly, we define a partial order on 4-valued valuations by pointwise comparisons of their atomic assignments: given 4-valued valuations ν and μ on *Args*, we denote by $\mu \geq_k \nu$ that $\mu(A) \geq_k \nu(A)$ for every $A \in Args$.
- A similar partial order is defined on 4-states labelings: we denote by $\leq_k$ the partial order on $\{\mathsf{in}, \mathsf{out}, \mathsf{none}, \mathsf{both}\}$ in which none is the $\leq_k$-minimal element, both is the $\leq_k$-maximal element, and in and out are (incomparable) intermediate elements. Accordingly, a partial order $\leq_k$ is defined on 4-states labeling by pointwise comparisons on the labels that they attach to the arguments: $lab_1 \geq_k lab_2$ iff $lab_1(A) \geq_k lab_2(A)$ for every $A \in Args$.

Note 52. Clearly, it holds that $\nu_1 \leq_k \nu_2$ iff $p\mathcal{VL}(\nu_1) \leq_k p\mathcal{VL}(\nu_2)$ and $lab_1 \leq_k lab_2$ iff $p\mathcal{LV}(lab_1) \leq_k p\mathcal{LV}(lab_2)$.

The partial orders $\leq_k$ defined above are known as the Belnap's knowledge orders on his 4-valued bilattice [25]. Intuitively, they reflect differences in the amount of information exhibited by the compared elements (see also [22]).

Lemma 53. *For every p-admissible labeling lab_a of an argumentation framework $\mathcal{AF}$ there is a p-complete labeling lab_c of $\mathcal{AF}$ such that $lab_c \geq_k lab_a$.*

Proof. Let lab_a be a p-admissible labeling of $\mathcal{AF} = \langle Args, Attack \rangle$. If it is also p-complete, we are done. Otherwise, lab_a violates one or more postulates among **pIn**$^+$, **pOut**$^+$, **pBoth**$^+$, **pNone**$^+$ for one or more arguments in *Args* (see Definition 20). On the other hand, lab_a is p-admissible, thus it satisfies postulates **pIn**, **pOut**, **pBoth** and **pNone**. Since the postulates regarding in-assignments and both-assignments of p-admissible and p-complete labelings coincide, the only postulates that may be violated are **pOut**$^+$ or **pNone**$^+$.

- Suppose first that lab_a violates **pNone**$^+$ for some argument A. In this case $lab_a(A) = \mathsf{none}$ and **pNone** is satisfied with respect to A. This may only happen if for every $B \in A^-$ it holds that $lab_a(B) = \mathsf{out}$. We therefore apply the following correction rule:

  [**none** $\rightsquigarrow$ **in**]: if $lab(A) = \mathsf{none}$ and $\forall B \in A^-$ it holds that $lab(B) = \mathsf{out}$, then let $lab(A) = \mathsf{in}$.

  The last rule fixes the problem regarding A (which now satisfies **pIn**$^+$), but it may cause a violation of **pNone**$^+$ regarding another argument: a none-labeled argument that was attacked by another none-labeled argument may now be attacked by an in-labeled argument. To fix this we need another rule:

  [**none** $\rightsquigarrow$ **out**]: if $lab(A) = \mathsf{none}$ and $\exists B \in A^-$ such that $lab(B) = \mathsf{in}$, then let $lab(A) = \mathsf{out}$.

  It is easy to verify that this additional rule indeed fixes the postulate violation and does not cause additional violations of the postulates for p-complete labelings.

- Suppose now that lab_a violates **pOut**$^+$ for some argument A. In this case $lab_a(A) = \mathsf{out}$ and **pOut** is satisfied with respect to A. This may only happen if for every $B \in A^-$ it holds that $lab_a(B) \in \{\mathsf{both}, \mathsf{out}\}$ (and at least one of them is assigned both). We therefore apply the following correction rule:

  [**out** $\rightsquigarrow$ **both**]: if $lab(A) = \mathsf{out}$ and $\forall B \in A^-$ it holds that $lab(B) \in \{\mathsf{both}, \mathsf{out}\}$, then let $lab(A) = \mathsf{both}$.

  Again, the last rule fixes the problem regarding A (which now satisfies **pBoth**$^+$), but it may cause a violation of **pIn**$^+$ regarding another argument: an in-labeled argument that was attacked only by out-labeled arguments may now be attacked by a both-labeled argument (and so **pIn**$^+$ is violated). To fix this we again need an additional rule:

  [**in** $\rightsquigarrow$ **both**]: if $lab(A) = \mathsf{in}$ and $\exists B \in A^-$ such that $lab(B) = \mathsf{both}$, then let $lab(A) = \mathsf{both}$.

  As in the previous case, it is easy to verify that this additional rule fixes the postulate violation and does not cause further violations of the postulates for p-complete labelings.

Let now lab_c be the labeling lab_a modified according to the above four correction rules. Then lab_c is a p-complete labeling of $\mathcal{AF}$, and since each rule increases the assignments with respect to the $\leq_k$-order (Definition 51), we have that $lab_c \geq_k lab_a$. □

Example 54. Consider again the p-admissible labelings of $\mathcal{AF}_1$ (Figure 1), listed in the Table of Example 22. Labeling number 2 in that table is not p-complete, since its assignment to argument C violates **pNone**$^+$. By correcting the labeling of C using the rule [**none** $\rightsquigarrow$ **in**] in the proof above and then correcting the labeling of D using the rule [**none** $\rightsquigarrow$ **out**] in the same proof, we get labeling number 4 in the same table, which *is* p-complete for $\mathcal{AF}_1$.

Lemma 55. *Let Const be a set of formulas in the language of* $\{\vee, \wedge, \neg\}$. *If* ν *is a 4-valued model of Const and* $\mu \geq_k \nu$, *then* μ *is a 4-valued model of Const as well.*

Proof. The above lemma resembles Lemma 35 in the 3-valued case. Again, its validity follows from the fact that the $\leq_k$ relation is extendable to complex formulas: if $\mu(A) \geq_k \nu(A)$ for every $A \in Args$ then $\mu(\psi) \geq_k \nu(\psi)$ for every formula ψ in the language of $\{\vee, \wedge, \neg\}$. (The proof here is, again, by induction on the structure of ψ). This is true, in particular, for every constraint ψ. Thus, if $\nu(\psi) \in \{t, \top\}$ for $\psi \in Const$, also $\mu(\psi) \in \{t, \top\}$, and so μ is a model of *Const*.[10] □

Proposition 56. *Every pCAF whose constraints are in the language of* $\{\vee, \wedge, \neg\}$ *has a p-complete extension/labeling.*

Proof. Let $p\,\mathcal{CAF} = \langle Args, Attack, Const\rangle$ be a p-constrained argumentation framework. By Proposition 50 there is a p-admissible labeling lab_p for $p\mathcal{CAF}$. In particular, lab_p is a p-admissible labeling of $\mathcal{AF} = \langle Attack, Const\rangle$ and $\mathcal{LV}(lab_p)$ is a model of *Const*. If lab_p is also a p-complete labeling of $\mathcal{AF}$, it is a p-complete labeling of $p\,\mathcal{CAF}$ and so we are done. Otherwise, if lab_p is *not* a p-complete labeling of $\mathcal{AF}$, by Lemma 53, there is a p-complete labeling lab_p^c of $\mathcal{AF}$ such that $lab_p^c \geq_k lab_p$. Moreover, since $p\mathcal{LV}(lab_p)$ is a model of *Const* we have, by Lemma 55, that $p\mathcal{LV}(lab_p^c)$ satisfies *Const*, and so lab_p^c is a complete labeling of $p\,\mathcal{CAF}$. □

Note 57. Examples 38 and 40 may be used for showing that also in the case of conflict-tolerant semantics when the implication $\supset$ appears in the constraints, the underlying pCAF may not have any p-complete extensions/labelings and that there may be several minimal p-complete extensions (with respect to the subset relation) for the same pCAF.

3.3. Conflict Minimization

Generally, although conflicts in pCAFs are sometimes unavoidable (e.g., when the set of constraints is not conflict-free, see Item (b) of Proposition 61 below), they obviously should be minimized as much as possible. In Example 48, for instance, the pCAF that consists of the argumentation framework of Figure 5 and the constraint $A \wedge B$ has three p-complete labelings. However, just two of these labelings are really informative, and the third one, assigning both to all the arguments, is somewhat anomalous. This motivates the next definition.

[10] Note that as in the 3-valued case, when $\supset$ is a connective in the language, a claim similar to that of Lemma 55 does not hold any longer. The same counter-example provided in Note 36 may be used here as well.

Definition 58.

- A p-admissible (respectively, p-complete) labeling *lab* of an argumentation framework $\mathcal{AF}$ is *minimally conflicting*, if there is no p-admissible (respectively, p-complete) labeling lab' of $\mathcal{AF}$, such that $\mathsf{Both}(lab') \subsetneq \mathsf{Both}(lab)$.
- We say that a p-admissible (respectively, p-complete) labeling *lab* of a pCAF $\langle Args, Attack, Const\rangle$ is *minimally conflicting*, if it is a minimally conflicting p-admissible (respectively, minimally conflicting p-complete) labeling of $\langle Args, Attack\rangle$.

Proposition 59. *The following two conditions are equivalent and define a minimally conflicting p-admissible (respectively, minimally conflicting p-complete) extension $\mathcal{E}$ of $\mathcal{AF}$:*

1. *There is no p-admissible (respectively, p-complete) extension $\mathcal{E}'$ of $\mathcal{AF}$, such that $\{A \mid A \in \mathcal{E}' \cap (\mathcal{E}')^+\} \subsetneq \{A \mid A \in \mathcal{E} \cap \mathcal{E}^+\}$.*
2. *$\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a minimally conflicting p-admissible (respectively, minimally conflicting p-complete) labeling of $\mathcal{AF}$.*

A similar equivalence holds for minimally conflicting p-extensions of a pCAF.

Proof. Straightforward from Proposition 21 and Definition 58. □

Example 60. Let us consider again the p-constrained argumentation framework $\mathcal{CAF}_1^{A\wedge B}$ of Example 47.

- Among the four p-admissible extensions of $\mathcal{CAF}_1^{A\wedge B}$, only $\{A, B\}$ is minimally conflicting. It corresponds to the labeling $\langle\{\}, \{C, D\}, \{A, B\}, \{\}\rangle$ that is minimally conflicting among the p-admissible labelings of $\mathcal{CAF}_1^{A\wedge B}$.
- The p-constrained framework $\mathcal{CAF}_\star^{A\wedge B}$, obtained from $\mathcal{CAF}_1^{A\wedge B}$ by removing the attack of B on D (and leaving everything else unchanged, including the constraint), has two minimally conflicting extensions: one, $\{A, B\}$, corresponds to the labeling $\langle\{\}, \{C\}, \{A, B\}, \{D\}\rangle$, and another one, $\{A, B, D\}$, corresponds to the labeling $\langle\{D\}, \{C\}, \{A, B\}, \{\}\rangle$. Both of these labelings are minimally conflicting among the p-admissible labelings of $\mathcal{CAF}_\star^{A\wedge B}$.

Minimally conflicting p-extensions reduce to a minimum the number of accepted arguments that are attacked by other accepted arguments. Two immediate consequences of this are considered in the next proposition.

Proposition 61.

a) All the minimally conflicting p-admissible extensions and the minimally conflicting p-complete extensions of a given argumentation framework are conflict-free.[11]

[11] Note, however, that the minimally conflicting sets among the *nonempty* p-complete extensions may not be conflict-free (see, e.g., Example 16).

b) *Let $p\mathcal{CAF}$ be a p-constrained framework for an argumentation framework $\mathcal{AF}$ and a set of constraints Const. Then Const is conflict-free iff all the minimally conflicting p-admissible extensions and the minimally conflicting p-complete extensions of $p\mathcal{CAF}$ are conflict-free.*

Proof. The first part follows from the fact that every argumentation framework has a complete (and so admissible) extension, thus it has a conflict-free p-complete (and p-admissible) extension. This implies that all of its minimally conflicting p-complete (and p-admissible) extensions must be conflict-free.

For the proof of the second part, note that if a minimally conflicting p-complete extension $\mathcal{E}$ of $p\mathcal{CAF}$ is conflict-free, then $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a conflict-free labeling of *Const* and $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ is a 3-valued model of *Const*, thus *Const* is conflict-free. Conversely, if *Const* is conflict-free, then since it is also p-admissible, it is in particular admissible, and so it has a model ν such that $p\mathcal{VL}(\nu)$ is an admissible labeling of $\mathcal{AF}$ and $p\mathcal{LE}(p\mathcal{VL}(\nu))$ is an admissible extension of $\mathcal{AF}$. The latter is extendable to a complete extension $\mathcal{E}$ of $\mathcal{AF}$. Now, $\mathcal{E}$ is a conflict-free p-complete (p-admissible) extension of $p\mathcal{CAF}$, and so, as in the proof of the first part, this implies that every minimally conflicting p-complete (p-admissible) extension of $p\mathcal{CAF}$ is conflict-free. $\square$

4. Reasoning with CAFs and pCAFs

In this section we show how the variety of semantics for CAFs and pCAFs considered previously in this paper can be represented (and computed) by propositional theories. In what follows we demonstrate this on pCAFs and 4-valued semantics. The case of CAFs and 3-valued semantics is obtained by some straightforward adjustments (which are often simplified representations).

Note 62. As the following table shows, under 4-valued semantics any state of mind regarding an argument A is expressible by formulas in $\mathcal{L}_{Args}$. In this table, we abbreviate formulas of the form $\psi \supset \mathsf{f}$ by $\mathsf{not}\ \psi$.

abbreviation	formula	satisfying assignments for A
$\mathsf{accept}(A)$	A	$t, \top$
$\mathsf{contradictory}(A)$	$\mathsf{accept}(A) \wedge \neg\mathsf{accept}(A)$	$\top$
$\mathsf{coherent}(A)$	$\mathsf{not\ contradictory}(A)$ [12]	$t, f, \bot$
$\mathsf{strong\text{-}accept}(A)$	$\mathsf{accept}(A) \wedge \mathsf{coherent}(A)$	t
$\mathsf{strong\text{-}reject}(A)$	$\neg\mathsf{accept}(A) \wedge \mathsf{coherent}(A)$	f
$\mathsf{undecided}(A)$	$\mathsf{not}\ (\mathsf{accept}(A) \vee \neg\mathsf{accept}(A))$	$\bot$

[12] Recall that this is an abbreviation of the formula $\mathsf{contradictory}(A) \supset \mathsf{f}$, i.e., $(A \wedge \neg A) \supset \mathsf{f}$.

Using the above notations, the postulates in Definition 20 for the p-admissible labelings of $\mathcal{AF} = \langle Args, Attack\rangle$ may be represented as follows:

$$
\begin{array}{ll}
\mathsf{pIn}(x): & \mathsf{strong\text{-}accept}(x) \supset \bigwedge_{y\in x^-} \mathsf{strong\text{-}reject}(y)\\
\mathsf{pOut}(x): & \mathsf{strong\text{-}reject}(x) \supset \bigvee_{y\in x^-} \mathsf{accept}(y)\\
\mathsf{pBoth}(x): & \mathsf{contradictory}(x) \supset\\
& \Big(\bigwedge_{y\in x^-}\Big(\mathsf{strong\text{-}reject}(y) \vee \mathsf{contradictory}(y)\Big) \wedge \bigvee_{y\in x^-} \mathsf{contradictory}(y)\Big)\\
\mathsf{pNone}(x): & \mathsf{undecided}(x) \supset \bigwedge_{y\in x^-}\Big(\mathsf{strong\text{-}reject}(y) \vee \mathsf{undecided}(y)\Big)
\end{array}
$$

Similarly, the p-complete labelings of $\mathcal{AF}$ may be represented as follows (Below, we abbreviate by $\psi \leftrightarrow \phi$ the formula $(\psi \supset \phi) \wedge (\phi \supset \psi)$):

$$
\begin{array}{ll}
\mathsf{pIn}^+(x): & \mathsf{strong\text{-}accept}(x) \leftrightarrow \bigwedge_{y\in x^-} \mathsf{strong\text{-}reject}(y)\\
\mathsf{pOut}^+(x): & \mathsf{strong\text{-}reject}(x) \leftrightarrow\\
& \Big(\bigvee_{y\in x^-} \mathsf{accept}(y) \wedge \bigvee_{y\in x^-}\Big(\mathsf{strong\text{-}accept}(y) \vee \mathsf{undecided}(y)\Big)\Big)\\
\mathsf{pBoth}^+(x): & \mathsf{contradictory}(x) \leftrightarrow\\
& \Big(\bigwedge_{y\in x^-}\Big(\mathsf{strong\text{-}reject}(y) \vee \mathsf{contradictory}(y)\Big) \wedge \bigvee_{y\in x^-} \mathsf{contradictory}(y)\Big)\\
\mathsf{pNone}^+(x): & \mathsf{undecided}(x) \leftrightarrow\\
& \Big(\bigwedge_{y\in x^-}\Big(\mathsf{strong\text{-}reject}(y) \vee \mathsf{undecided}(y)\Big) \wedge \bigvee_{y\in x^-} \mathsf{undecided}(y)\Big)
\end{array}
$$

Clearly, an expression $\Phi(x)$ of those described above becomes a meaningful formula (in $\mathcal{L}_{Args}$) only when, given an argumentation framework $\mathcal{AF} = \langle Args, Attack\rangle$, its variable x is substituted by an atom A that is associated with an argument $A \in Args$ and the elements in A^- are determined by $Attack$. In what follows we denote by $\Psi(A, \mathcal{AF})$ the formula that is obtained from $\Phi(x)$ in this way.

Example 63. Consider the argumentation framework $\mathcal{AF}$ in Figure 5 (Example 48). When $x = C$ we have that $x^- = \{A, B\}$, and so:

$$\mathsf{pIn}(C, \mathcal{AF}) \;=\; \mathsf{strong\text{-}accept}(C) \supset (\mathsf{strong\text{-}reject}(A) \wedge \mathsf{strong\text{-}reject}(B)).$$

A 4-valued model of $\mathsf{pIn}(C, \mathcal{AF})$ that assigns t to C must assign f to both of A and B. Thus, intuitively, this formula requires that every p-admissible labeling of $\mathcal{AF}$ that (strongly) accepts C must (strongly) reject both of A and B (see also Proposition 64 below).

For representing the p-labelings for $\mathcal{AF}$ we use the following theories:

$$\begin{aligned}\mathsf{pADM}(\mathcal{AF}) = \; & \bigcup_{x \in Args} \mathsf{pIn}(x, \mathcal{AF}) \cup \bigcup_{x \in Args} \mathsf{pOut}(x, \mathcal{AF}) \cup \\ & \bigcup_{x \in Args} \mathsf{pBoth}(x, \mathcal{AF}) \cup \bigcup_{x \in Args} \mathsf{pNone}(x, \mathcal{AF})\end{aligned}$$

$$\begin{aligned}\mathsf{pCMP}(\mathcal{AF}) = \; & \bigcup_{x \in Args} \mathsf{pIn}^+(x, \mathcal{AF}) \cup \bigcup_{x \in Args} \mathsf{pOut}^+(x, \mathcal{AF}) \cup \\ & \bigcup_{x \in Args} \mathsf{pBoth}^+(x, \mathcal{AF}) \cup \bigcup_{x \in Args} \mathsf{pNone}^+(x, \mathcal{AF})\end{aligned}$$

The theories that represent p-labelings of pCAFs are obtained by adding the set of constraints to the above theories. For $p\,\mathcal{CAF} = \langle Args, Attack, Const \rangle$ with $\mathcal{AF} = \langle Args, Attack \rangle$, we define:

$$\mathsf{pADM}(p\,\mathcal{CAF}) = \mathsf{pADM}(\mathcal{AF}) \cup Const$$

$$\mathsf{pCMP}(p\,\mathcal{CAF}) = \mathsf{pCMP}(\mathcal{AF}) \cup Const$$

Proposition 64. *Let $\mathcal{AF} = \langle Args, Attack \rangle$ be an argumentation framework. Then:*

- *There is a one-to-one correspondence between the 4-valued models of the theory $\mathsf{pADM}(\mathcal{AF})$, the 4-states p-admissible labelings of $\mathcal{AF}$, and the p-admissible extensions of $\mathcal{AF}$. Moreover, it holds that*
  - *if ν is a model of $\mathsf{pADM}(\mathcal{AF})$ then $p\mathcal{VL}(\nu)$ is a p-admissible labeling of $\mathcal{AF}$ and $p\mathcal{LE}(p\mathcal{VL}(\nu))$ is a p-admissible extension of $\mathcal{AF}$.*
  - *If lab is a p-admissible labeling of $\mathcal{AF}$ then $p\mathcal{LV}(lab)$ is a model of $\mathsf{pADM}(\mathcal{AF})$ and $p\mathcal{LE}(lab)$ is a p-admissible extension of $\mathcal{AF}$.*
  - *If $\mathcal{E}$ is a p-admissible extension of $\mathcal{AF}$ then $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ is a model of $\mathsf{pADM}(\mathcal{AF})$ and $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-admissible labeling of $\mathcal{AF}$.*
- *There is a one-to-one correspondence between the 4-valued models of the theory $\mathsf{pCMP}(\mathcal{AF})$, the 4-states p-complete labelings of $\mathcal{AF}$, and the p-complete extensions of $\mathcal{AF}$. Moreover, it holds that*
  - *if ν is a model of $\mathsf{pCMP}(\mathcal{AF})$ then $p\mathcal{VL}(\nu)$ is a p-complete labeling of $\mathcal{AF}$ and $p\mathcal{LE}(p\mathcal{VL}(\nu))$ is a p-complete extension of $\mathcal{AF}$.*
  - *If lab is a p-complete labeling of $\mathcal{AF}$ then $p\mathcal{LV}(lab)$ is a model of $\mathsf{pCMP}(\mathcal{AF})$ and $p\mathcal{LE}(lab)$ is a p-complete extension of $\mathcal{AF}$.*
  - *If $\mathcal{E}$ is a p-complete extension of $\mathcal{AF}$ then $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ is a model of $\mathsf{pCMP}(\mathcal{AF})$ and $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-complete labeling of $\mathcal{AF}$.*

Proof. We show the first item; the proof of the second item is similar.
The one-to-one correspondence between the p-admissible extensions and the p-admissible labelings of $\mathcal{AF}$ is shown in the first item of Proposition 21. It therefore remains to show the correspondence between 4-valued models of $\mathsf{pADM}(\mathcal{AF})$ and the 4-states p-admissible labelings of $\mathcal{AF}$. Indeed,

- Let ν be a model of $\mathsf{pADM}(\mathcal{AF})$ and suppose that $\nu(A) = t$. Then $\nu(\mathsf{strong\text{-}accept}(A)) = t$ and since ν satisfies $\mathsf{pIn}(A, \mathcal{AF})$, it holds that for every $B \in A^-$, $\nu(\mathsf{strong\text{-}reject}(B)) \in \{t, \top\}$, Thus, for every $B \in A^-$, $\nu(B) = f$. It follows that for every argument A such that $p\mathcal{VL}(\nu)(A) = \mathsf{in}$, it holds that $p\mathcal{VL}(\nu)(B) = \mathsf{out}$ whenever $B \in A^-$. Hence $p\mathcal{VL}(\nu)$ satisfies the postulate **pIn**. Similar considerations show that the fact that ν satisfies the formulas $\mathsf{pOut}(x, \mathcal{AF})$, $\mathsf{pBoth}(x, \mathcal{AF})$, and $\mathsf{pNone}(x, \mathcal{AF})$ for every $x \in Args$ guarantees, respectively, that $p\mathcal{VL}(\nu)$ satisfies the postulates **pOut**, **pBoth** and **pNone**. Thus $p\mathcal{VL}(\nu)$ is a p-admissible labeling of $\mathcal{AF}$.

- Let lab be a p-admissible labeling of $\mathcal{AF}$ such that $lab(A) = \mathsf{in}$. Then $p\mathcal{LV}(lab)(A) = t$, and so we have that $p\mathcal{LV}(lab)(\mathsf{strong\text{-}reject}(A)) = f$, $p\mathcal{LV}(lab)(\mathsf{contradictory}(A)) = f$, and $p\mathcal{LV}(lab)(\mathsf{undecided}(A)) = f$. This implies, respectively, that $p\mathcal{LV}(lab)$ satisfies $\mathsf{pOut}(A, \mathcal{AF})$, $\mathsf{pBoth}(A, \mathcal{AF})$, and $\mathsf{pNone}(A, \mathcal{AF})$. The fact that $p\mathcal{LV}(lab)$ satisfies also $\mathsf{pIn}(A, \mathcal{AF})$ follows from the fact that lab satisfies the postulate **pIn** and so $lab(B) = \mathsf{out}$ for every $B \in A^-$. This implies that $p\mathcal{LV}(lab)(B) = f$ for every $B \in A^-$, and so $p\mathcal{LV}(lab)(\mathsf{strong\text{-}reject}(B)) = t$ for every such B.
  The cases in which $lab(A) \in \{\mathsf{out}, \mathsf{both}, \mathsf{none}\}$ are similar, and so $p\mathcal{LV}(lab)$ is indeed a model of $\mathsf{pADM}(\mathcal{AF})$. $\square$

A similar proposition holds also for pCAFs:

Proposition 65. *Let $p\,\mathcal{CAF} = \langle Args, Attack, Const \rangle$ be a p-constrained argumentation framework. Then:*

- *There is a one-to-one correspondence between the 4-valued models of the theory $\mathsf{pADM}(p\,\mathcal{CAF})$, the 4-states p-admissible labelings of $p\,\mathcal{CAF}$, and the p-admissible extensions of $p\,\mathcal{CAF}$. Moreover, it holds that*
  - *if ν is a model of $\mathsf{pADM}(p\mathcal{CAF})$ then $p\mathcal{VL}(\nu)$ is a p-admissible labeling of $p\mathcal{CAF}$ and $p\mathcal{LE}(p\mathcal{VL}(\nu))$ is a p-admissible extension of $p\mathcal{CAF}$.*
  - *If lab is a p-admissible labeling of $p\mathcal{CAF}$ then $p\mathcal{LV}(lab)$ is a model of $\mathsf{pADM}(p\,\mathcal{CAF})$ and $p\mathcal{LE}(lab)$ is a p-admissible extension of $p\,\mathcal{CAF}$.*
  - *If $\mathcal{E}$ is a p-admissible extension of $p\,\mathcal{CAF}$ then $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ is a model of $\mathsf{pADM}(p\,\mathcal{CAF})$ and $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-admissible labeling of $p\,\mathcal{CAF}$.*

- *There is a one-to-one correspondence between the 4-valued models of the theory $\mathsf{pCMP}(p\mathcal{CAF})$, the 4-states p-complete labelings of $p\mathcal{CAF}$, and the p-complete extensions of $p\,\mathcal{CAF}$. Moreover, it holds that*
  - *if ν is a model of $\mathsf{pCMP}(p\mathcal{CAF})$ then $p\mathcal{VL}(\nu)$ is a p-complete labeling of $p\,\mathcal{CAF}$ and $p\mathcal{LE}(p\mathcal{VL}(\nu))$ is a p-complete extension of $p\,\mathcal{CAF}$.*
  - *If lab is a p-complete labeling of $p\,\mathcal{CAF}$ then $p\mathcal{LV}(lab)$ is a model of $\mathsf{pCMP}(p\,\mathcal{CAF})$ and $p\mathcal{LE}(lab)$ is a p-complete extension of $p\,\mathcal{CAF}$.*

– *If $\mathcal{E}$ is a p-complete extension of $p\,\mathcal{CAF}$ then $p\mathcal{LV}(p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E}))$ is a model of* $\mathsf{pCMP}(p\,\mathcal{CAF})$ *and $p\mathcal{EL}_{\mathcal{AF}}(\mathcal{E})$ is a p-complete labeling of $p\,\mathcal{CAF}$.*

Proof. Similar to that of Proposition 64. □

Example 66. Let $p\,\mathcal{CAF}$ be the p-constrained argumentation framework considered in Example 48: the argumentation framework is shown in Figure 5 and the constraint is $\mathsf{accept}(A) \wedge \mathsf{accept}(B)$. The theory $\mathsf{pADM}(p\,\mathcal{CAF})$, simplified by some standard rewriting rules, is shown in Figure 6.

$$\begin{array}{l}
\mathsf{strong\text{-}accept}(A) \supset \mathsf{strong\text{-}reject}(B)\\
\mathsf{strong\text{-}accept}(B) \supset \mathsf{strong\text{-}reject}(A)\\
\mathsf{strong\text{-}accept}(C) \supset (\mathsf{strong\text{-}reject}(A) \wedge \mathsf{strong\text{-}reject}(B))\\
\mathsf{strong\text{-}accept}(D) \supset \mathsf{strong\text{-}reject}(C)\\
\\
\mathsf{strong\text{-}reject}(A) \supset \mathsf{accept}(B)\\
\mathsf{strong\text{-}reject}(B) \supset \mathsf{accept}(A)\\
\mathsf{strong\text{-}reject}(C) \supset (\mathsf{accept}(A) \vee \mathsf{accept}(B))\\
\mathsf{strong\text{-}reject}(D) \supset \mathsf{accept}(C)\\
\\
\mathsf{contradictory}(A) \supset \mathsf{contradictory}(B)\\
\mathsf{contradictory}(B) \supset \mathsf{contradictory}(A)\\
\mathsf{contradictory}(C) \supset (((\mathsf{strong\text{-}reject}(A) \vee \mathsf{contradictory}(A))\\
\qquad\qquad\qquad\qquad \wedge\ (\mathsf{strong\text{-}reject}(B) \vee \mathsf{contradictory}(B)))\\
\qquad\qquad\qquad\qquad \wedge\ (\mathsf{contradictory}(A) \vee \mathsf{contradictory}(B)))\\
\mathsf{contradictory}(D) \supset \mathsf{contradictory}(C)\\
\\
\mathsf{undecided}(A) \supset (\mathsf{strong\text{-}reject}(B) \vee \mathsf{undecided}(B))\\
\mathsf{undecided}(B) \supset (\mathsf{strong\text{-}reject}(A) \vee \mathsf{undecided}(A))\\
\mathsf{undecided}(C) \supset (((\mathsf{strong\text{-}reject}(A) \vee \mathsf{undecided}(A))\\
\qquad\qquad\qquad\qquad \wedge\ (\mathsf{strong\text{-}reject}(B) \vee \mathsf{undecided}(B)))\\
\mathsf{undecided}(D) \supset (\mathsf{strong\text{-}reject}(C) \vee \mathsf{undecided}(C))\\
\\
\mathsf{accept}(A) \wedge \mathsf{accept}(B)
\end{array}$$

Figure 6: The theory $\mathsf{pADM}(p\,\mathcal{CAF})$ of Example 66

The three models of $\mathsf{pADM}(p\,\mathcal{CAF})$ are listed in the table below.

	A	B	C	D
1	$\top$	$\top$	$\top$	$\top$
2	$\top$	$\top$	f	t
3	$\top$	$\top$	f	$\bot$

As guaranteed by Proposition 65, these models correspond to the three p-admissible lablings of $p\,\mathcal{CAF}$ (see Example 48).

Note 67. Propositional theories for reasoning with p-labeling and p-extensions of (p-constrained) argumentation frameworks are given also in [17] and [18]. Similar theories for reasoning with conflict-free semantics are described in [29]. The main difference is that in these papers the underlying semantics is two-valued and the shift back and forth from and to four-valued semantics is done through syntactical mappings using signed formulas (see also [30]). In our case everything remains within the four-valued context.

5. Conclusion

The incorporation of integrity constraints in argumentation frameworks is a useful way of providing information about arguments. Such information may involve, for instance, meta-data in the form of preferences among arguments, external knowledge about the domain of discourse, or some instructive information that simplifies and clarifies the intended semantics at hand.

In this paper we extended and improved several previous works on constraining argumentation frameworks: the conflict-free semantics for CAFs considered in [24] are extended to conflict-tolerant ones, allowing to handle situations which are implicitly inconsistent or cases where the constraints themselves are contradictory. Another difference from the treatment in [24] is that here the constraints are evaluated with respect to the same semantics as that of the arguments: three-valued semantics for conflict-free systems and four-valued semantics for conflict-tolerant systems. In addition, the discussion on constrained argumentation frameworks in [18], aimed at demonstrating the usefulness of conflict-tolerant semantics for argumentation frameworks, is largely extended in our case. In particular, the restriction of using only atomic constraints is lifted, and more general results on the existence of complete extensions for different forms of constraints are provided.

We note, finally, that evaluating the usefulness of constrained argumentation frameworks and assessing the plausibility of their semantics in realistic situations require a substantial experimental study. This remains a subject for future work.

References

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

[2] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81–132.

[3] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: Proc. ICLP'88, MIT Press, 1988, pp. 1070–1080.

[4] Y. Wu, M. Caminada, D. M. Gabbay, Complete extensions in argumentation coincide with 3-valued stable models in logic programming, Studia Logica 93 (2009) 383–403.

[5] A. van Gelder, K. A. Ross, J. S. Schlipf, The well-founded semantics for general logic programs., Journal of the ACM 38 (1991) 620–650.

[6] G. Governatori, M. J. Maher, G. Antoniou, D. Billington, Argumentation semantics for defeasible logic, Journal of Logic and Computation 14 (2004) 675–702.

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

[8] T. J. M. Bench-Capon, Persuasion in practical argument using value-based argumentation frameworks, Journal of Logic and Computation 13 (2003) 429–448.

[9] S. Modgil, Reasoning about preferences in argumentation frameworks, Artificial Intelligence 173 (2009) 901–934.

[10] H. Prakken, AI & law, logic and argument schemes, Argumentation 19 (2005) 303–320.

[11] R. Booth, S. Kaci, T. Rienstra, L. van der Torre, A logical theory about dynamics in abstract argumentation, in: W. Liu, V. S. Subrahmanian, J. Wijsen (Eds.), Proc. SUM 2013, volume 8078 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 148–161.

[12] C. Cayrol, F. D. de Saint-Cyr, M. Lagasquie-Schiex, Change in abstract argumentation frameworks: adding an argument, Artificial Intelligent Research 38 (2010) 49–84.

[13] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The Knowledge Engineering Review 26 (2011) 365–410.

[14] P. Baroni, M. Giacomin, Semantics for abstract argumentation systems, in: I. Rahwan, G. R. Simary (Eds.), Argumentation in Artificial Intelligence, Springer, 2009, pp. 25–44.

[15] M. Caminada, On the issue of reinstatement in argumentation, in: M. Fischer, W. van der Hoek, B. Konev, A. Lisitsa (Eds.), Proc. JELIA'06, volume 4160 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 111–123.

[16] M. Caminada, D. M. Gabbay, A logical account of formal argumentation, Studia Logica 93 (2009) 109–145. Special issue: New ideas in argumentation theory.

[17] O. Arieli, Conflict-tolerant semantics for argumentation frameworks, in: L. Fariñas del Cerro, A. Herzig, J. Mengin (Eds.), Proc. JELIA'12, volume 7519 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 28–40.

[18] O. Arieli, On the acceptance of loops in argumentation frameworks, Journal of Logic and Computation (2014). Accepted, doi: 10.1093/logcom/exu009.

[19] S. C. Kleene, Introduction to Metamathematics, Van Nostrand, 1950.

[20] A. Avron, Natural 3-valued logics: Characterization and proof theory, Journal of Symbolic Logic 56 (1991) 276–294.

[21] J. Słupecki, Der volle dreiwertige aussagenkalkül, Com. Rend. Soc. Sci. Lett. de Varsovie 29 (1936) 9–11.

[22] O. Arieli, A. Avron, The value of the four values, Artificial Intelligence 102 (1998) 97–141.

[23] O. Arieli, Towards constraints handling by conflict tolerance in abstract argumentation frameworks., in: C. Boonthem-Denecke, G. M. Youngblood (Eds.), Proc. FLAIRS'13, AAAI Press, 2013, pp. 585–590.

[24] S. Coste-Marquis, C. Devred, P. Marquis, Constrained argumentation frameworks, in: P. Doherty, J. Mylopoulos, C. A. Welty (Eds.), Proc. KR'06, AAAI Press, 2006, pp. 112–122.

[25] N. D. Belnap, A useful four-valued logic, in: J. M. Dunn, G. Epstein (Eds.), Modern Uses of Multiple-Valued Logics, Reidel Publishing Company, 1977, pp. 7–37.

[26] N. C. A. da Costa, On the theory of inconsistent formal systems, Notre Dame Journal of Formal Logic 15 (1974) 497–510.

[27] I. D'Ottaviano, N. C. da Costa, Sur un problèm de Jakowśki, C. R. Acad Sc. Paris, Volume 270, Sèrie A (1970) 1349–1353.

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

[29] O. Arieli, M. Caminada, A QBF-based formalization of abstract argumentation semantics, Journal of Applied Logic 11 (2013) 229–252.

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