Argument & Computation 11 (2020) 305–339
DOI 10.3233/AAC-200520
IOS Press

A principle-based robustness analysis of admissibility-based argumentation semantics

Tjitze Rienstra [a,*], Chiaki Sakama [b], Leendert van der Torre [c] and Beishui Liao [d]
[a] *Institute for Web Science and Technologies, University of Koblenz-Landau, Germany*
E-mail: rienstra@uni-koblenz.de
[b] *Department of Systems Engineering, Wakayama University, Wakayama, Japan*
E-mail: sakama@sys.wakayama-u.ac.jp
[c] *Department of Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg*
E-mail: leon.vandertorre@uni.lu
[d] *Department of Philosophy, Zhejiang University, Hangzhou, China*
E-mail: baiseliao@zju.edu.cn

Abstract. The principle-based approach is a methodology to classify and analyse argumentation semantics. In this paper we classify seven of the main alternatives for argumentation semantics using a set of new *robustness* principles. These principles complement Baroni and Giacomin's original classification and deal with the behaviour of a semantics when the argumentation framework changes due to the addition or removal of an attack between two arguments. We distinguish so-called *persistence* principles and *monotonicity* principles, where the former deal with the question of whether a labelling or extension of an argumentation framework under a given semantics persists after a change, and the latter with the question of whether new labellings or extensions are created after a change. We furthermore show in which sense labelling-based and extension-based semantics lead to subtly different principles and results. Our results can be used for choosing a semantics for a particular application, or to guide the search for new argumentation semantics, but they have been used also in the design of algorithms.

Keywords: Artificial intelligence, knowledge representation and reasoning, formal argumentation, abstract argumentation, principle-based analysis

1. Introduction

Many semantics have been proposed in the literature on abstract argumentation. One way to compare and evaluate them is to look at examples of argumentation frameworks that bring to light important differences between semantics or highlight a particular strength or weakness of a semantics. While such an *example-based* approach may produce valuable insights, a deeper understanding of the differences and similarities of semantics requires a more systematic study of their properties. Baroni *et al.* introduced the so called *principle-based* approach [4,60], which takes as a starting point principles – that may or may not be satisfied by a semantics – representing various intuitions and requirements, which are used to evaluate a semantics or to guide the search for new semantics. These principles range from basic properties of extensions such as conflict-freeness and admissibility to principles that reflect desirable behaviour

*Corresponding author: Tjitze Rienstra, University of Koblenz-Landau, Universitätsstraße 1, 56070, Koblenz. Tel.: +49 261 287-2779; E-mail: rienstra@uni-koblenz.de.

in certain classes of examples, such as crash-resistance, non-interference and directionality [4,30]. Semantics for argumentation may also be characterised in terms of *strong equivalence* conditions, which identify the syntactical differences that are semantically indistinguishable under a given semantics [7–10,18,54], or in terms of their *signature*, which identifies the set of all possible sets of extensions or labellings that can be expressed under a given semantics [42,44,47]. The general approach of focusing on general properties in the evaluation of a semantics has also been applied to ranking-based semantics [2,19,24] and instantiated argumentation formalisms [27,29].

In this paper we extend the principle-based investigation of semantics for abstract argumentation and study principles that characterise the *robustness* of extensions or labellings under a semantics with respect to the removal or addition of attacks. These principles are relative to the status assigned to the two arguments between which the attack is added or removed. The set of different statuses we can distinguish depends on whether we look at labelling-based or extension-based semantics. A labelling-based semantics σ maps every argumentation framework to a set of labellings, which are functions that assign to each argument a label **I** (or accepted), **O** (or rejected) or **U** (or undecided) [28]. Given a labelling-based semantics σ we define, for each combination X and Y of labels, the following four principles:

- XY **addition persistence**: a σ labelling of an argumentation framework F in which x is labelled X and y is labelled Y is still a σ labelling of F after adding an attack from x to y.
- XY **removal persistence**: a σ labelling of an argumentation framework F in which x is labelled X and y is labelled Y is still a σ labelling of F if removing the attack from x to y.
- XY **addition monotonicity**: if in all σ labellings of an argumentation framework F, x is labelled X and y is labelled Y, then adding an attack from x to y does not lead to new σ labellings.
- XY **removal monotonicity**: if in all σ labellings of an argumentation framework F, x is labelled X and y is labelled Y, then removing an attack from x to y does not lead to new σ labellings.

For the extension-based setting we define the same set of principles except that the status of an argument is either *in an extension* (or, equivalently, labelled **I**) or *not in an extension* (or, equivalently, labelled either **O** or **U**). Thus, in the extension-based setting we do not have the ability to distinguish rejected and undecided arguments. The extension-based principles therefore have less discriminative power than the labelling-based ones. Nevertheless, the results obtained in the extension-based setting do provide additional insights. In particular, the extension-based setting leads to weaker forms of addition and removal monotonicity than in the labelling-based setting. That is, even if a semantics σ violates all variants of addition/removal monotonicity in the labelling-based setting, it may still satisfy variants of addition/removal under the extension-based setting.

We investigate the principles discussed here with respect to all the admissibility-based semantics that have been defined in the literature. Thus, we focus on the complete, grounded, preferred and stable semantics, which were defined by Dung in terms of extensions [40] and by Caminada in terms of labellings [28], as well as the semi-stable [30], ideal [41] and eager [26] semantics. The investigation of non-admissible semantics is planned for a future publication.

The results we obtain provide insight into the robustness of the evaluation of an argumentation framework under a given semantics when the argumentation framework changes. While we do not argue that any of the principles we investigate must necessarily be satisfied by a semantics in general, some principles can be regarded as reflecting reasonable or desirable requirements. Let us demonstrate this point by considering **OO** addition persistence. This principle reflects the intuition that a point of view on argument acceptance represented by a labelling L still represents a valid point of view after adding an attack from x to y, if both x and y are rejected in L. This can be regarded as reasonable because the added attack does not introduce a conflict with respect to L and does not change the justification for the rejection of x and y. Thus, the added attack provides no explicit reason to give up on L as a reasonable point of view on argument acceptance. Indeed, this is true under the complete semantics, which satisfies **OO** addition persistence. The complete semantics similarly satisfies **OU**, **UO**, **UU**, **OI** and **IO** addition persistence, as well as **OI**, **OU**, **UO** and **OO** removal persistence, where the removal of the attack does not invalidate the justification of the label of the second argument. Our results show, however, that not all the semantics we consider satisfy these principles, even though all of them associate an argumentation framework with a set of complete labellings. Other principles can similarly be regarded as undesirable. For instance, **II** addition persistence reflects the intuition that a point of view on argument acceptance survives even if we add an attack between two arguments that are both accepted. This amounts to permitting points of view on argument acceptance that contain conflicts, which is not desirable unless one considers a conflict-tolerant semantics [3]. In general, whether a principle is desirable depends on how one expects a semantics to behave with the argumentation framework changes, and this depends on the application. The results we obtain can therefore be used to determine whether a semantics is suitable for a given application.

The principles we investigate are also useful in the design of algorithms. For example, Niskanen et al. [51] use persistence principles in the design of an algorithm for computing extensions of incomplete argumentation frameworks, where one can specify that attacks between certain arguments may or may not exist. Persistence and monotonicity principles are also useful in addressing *enforcement* problems in abstract argumentation [11,12,25,34,53]. This is the problem of determining minimal sets of changes to an argumentation framework in order to enforce some result, such as the acceptance of a given set of arguments. Because persistence and monotonicity principles can be used to determine which changes to the attack relation of an argumentation framework do or do not change its evaluation, these principles can be used to guide the search for sets of changes in the enforcement problem. This idea has already been used for extension enforcement under the grounded semantics [53].

This paper extends results presented earlier [56], which focused on addition and removal persistence as well as addition monotonicity, but did not consider removal monotonicity and did not include results for the ideal and eager semantics. In the current paper we additionally examine persistence and monotonicity principles from the extension-based perspective, where we distinguish arguments that are accepted from those that are not, but where no distinction is made between rejected and undecided arguments. In case of addition and removal monotonicity, this different perspective yields a set of principles that are distinct from the labelling-based ones.

The layout of this paper is as follows. In Section 2 we recall the necessary definitions concerning abstract argumentation theory. Then we first focus on labelling-based semantics. The addition and removal persistence principles are discussed in Section 3 and 4, while addition and removal monotonicity principles are discussed in Section 5 and 6. We then discuss, in Section 7, persistence and monotonicity principles for extension-based semantics, and summarise and contrast our results in Section 8. We discuss directions for future work in Section 9, survey related work in Section 10, and conclude in Section 11.

2. Preliminaries

An *argumentation framework* is a directed graph represented by a set A of *arguments* and a binary relation $\rightsquigarrow$ over A called the *attack relation* [40]. We assume throughout this paper that a set $\mathcal{U}$, called the *universe of arguments* and whose elements are called *arguments*, is given, and that the set of arguments in an argumentation framework is finite.

Definition 1. Let $\mathcal{U}$ be a set called the *universe of arguments*. An *argumentation framework* is a pair $F = (A, \rightsquigarrow)$ where A is a finite subset of $\mathcal{U}$ and $\rightsquigarrow$ is a binary relation over A called the *attack relation*. We denote by $\mathcal{F}$ the set of all argumentation frameworks.

Given an argumentation framework $(A, \rightsquigarrow)$ we say that an argument $a \in A$ *attacks* an argument $b \in A$ if and only if $(a, b) \in \rightsquigarrow$. We will mostly use infix notation and write $a \rightsquigarrow b$ instead of $(a, b) \in \rightsquigarrow$. Given an argumentation framework $(A, \rightsquigarrow)$ and an argument $x \in A$ we denote by x^- the set of arguments attacking x and by x^+ the set of arguments attacked by x. Given a set $B \subseteq A$ we denote by B^- the set of arguments attacking some $x \in B$ and by B^+ the set of arguments attacked by some $x \in B$.

A *labelling-based* semantics maps every argumentation framework to a set of *labellings*, which are functions that map every argument of an argumentation framework to a label. All the labelling-based semantics considered in this paper are defined using three possible labels: **I** indicates that the argument is accepted, **O** that the argument is rejected, and **U** that the acceptance of the argument is undecided [28].

Definition 2. A *labelling* of an argumentation framework $F = (A, \rightsquigarrow)$ is a function $L : A \to \{\mathbf{I}, \mathbf{O}, \mathbf{U}\}$. Given a label $l \in \{\mathbf{I}, \mathbf{O}, \mathbf{U}\}$ we define $L^{-1}(l)$ as $\{x \in A \mid L(x) = l\}$. We denote by $\mathcal{L}(F)$ the set of all labellings of F. We also denote a labelling L by the set of pairs $\{(x_1, L(x_1)), \ldots, (x_n, L(x_n))\}$ where $A = \{x_1, \ldots, x_n\}$.

Definition 3. A *labelling-based semantics* σ defines a function $\mathcal{L}_\sigma$ that associates every $F \in \mathcal{F}$ with a set $\mathcal{L}_\sigma(F) \subseteq \mathcal{L}(F)$.

A *complete* labelling is a labelling in which an argument is accepted if and only if all attackers are rejected, and an argument is rejected if and only if some attacker is accepted. All the semantics we consider in this paper are based on complete labellings.

Definition 4. Let $F = (A, \rightsquigarrow)$ be an argumentation framework. A labelling $L \in \mathcal{L}(F)$ is *complete* if and only if, for all $x \in A$:

- $L(x) = \mathbf{I}$ if and only if, for all $y \in x^-$, $L(y) = \mathbf{O}$.
- $L(x) = \mathbf{O}$ if and only if, for some $y \in x^-$, $L(y) = \mathbf{I}$.

Given two labelling L, L', we say that L' is at least as committed as L (written $L' \sqsubseteq L$) if every argument that is labelled **I** (resp. **O**) by L' is also labelled **I** (resp. **O**) by L.

Definition 5. Let $F = (A, \rightsquigarrow)$ be an argumentation framework. The *committedness* relation $\sqsubseteq \subseteq \mathcal{L}(F) \times \mathcal{L}(F)$ is defined by $L \sqsubseteq L'$ if and only if $L^{-1}(\mathbf{I}) \subseteq L'^{-1}(\mathbf{I})$ and $L^{-1}(\mathbf{O}) \subseteq L'^{-1}(\mathbf{O})$.

We furthermore write $L \sqsubset L'$ as shorthand for $L \sqsubseteq L' \wedge L \not\sqsupseteq L'$. Given a set M of labellings we denote by $\sqcap M$ the labelling in which an argument x is labelled **I** (resp. **O**) if the argument is labelled **I** (resp. **O**) by all elements of M, and x is labelled **U** otherwise. Intuitively, $\sqcap M$ represents the "consensus" among M in terms of the arguments that are accepted and rejected.

Definition 6. Given an argumentation framework $F = (A, \rightsquigarrow)$ and a set $M \subseteq \mathcal{L}(F)$ we define $\sqcap M$ by $\sqcap M = L$, where L is the labelling of F defined by

(1) $L(x) = \mathbf{I}$ if and only if $\forall L' \in M$, $L(x) = \mathbf{I}$, and

(2) $L(x) = \mathbf{O}$ if and only if $\forall L' \in M$, $L'(x) = \mathbf{O}$.

The following definition introduces the labelling-based *complete* (*co*), *grounded* (*gr*), *preferred* (*pr*), *stable* (*stb*), *semi-stable* (*ss*), *ideal* (*id*), and *eager* (*ea*) semantics [26,28,30,41].

Definition 7. Let $F = (A, \rightsquigarrow)$ be an argumentation framework.

- $\mathcal{L}_{co}(F) = \{L \in \mathcal{L}(F) \mid L$ is a complete labelling of $F\}$
- $\mathcal{L}_{gr}(F) = \{L \in \mathcal{L}_{co}(F) \mid \nexists L' \in \mathcal{L}_{co}(F)$ such that $L'^{-1}(\mathbf{I}) \subset L^{-1}(\mathbf{I})\}$
- $\mathcal{L}_{pr}(F) = \{L \in \mathcal{L}_{co}(F) \mid \nexists L' \in \mathcal{L}_{co}(F)$ such that $L^{-1}(\mathbf{I}) \subset L'^{-1}(\mathbf{I})\}$
- $\mathcal{L}_{ss}(F) = \{L \in \mathcal{L}_{co}(F) \mid \nexists L' \in \mathcal{L}_{co}(F)$ such that $L'^{-1}(\mathbf{U}) \subset L^{-1}(\mathbf{U})\}$
- $\mathcal{L}_{stb}(F) = \{L \in \mathcal{L}_{co}(F) \mid L^{-1}(\mathbf{U}) = \emptyset\}$
- $\mathcal{L}_{id}(F) = \{L \in \mathcal{L}_{co}(F) \mid L \sqsubseteq \sqcap\mathcal{L}_{pr}(F)$ and $\nexists L' \in \mathcal{L}_{co}(F)$ such that $L' \sqsubseteq \sqcap\mathcal{L}_{pr}(F)$ and $L \sqsubset L'\}$
- $\mathcal{L}_{ea}(F) = \{L \in \mathcal{L}_{co}(F) \mid L \sqsubseteq \sqcap\mathcal{L}_{ss}(F)$ and $\nexists L' \in \mathcal{L}_{co}(F)$ such that $L' \sqsubseteq \sqcap\mathcal{L}_{ss}(F)$ and $L \sqsubset L'\}$

We now state two well-known facts about the semantics defined in Definition 7 that will be used throughout this paper. First of all, under all semantics except for the stable semantics, every argumentation framework has at least one labelling.

Proposition 1. *Let F be an argumentation framework and $\sigma \in \{co, gr, pr, ss, id, ea\}$. Then $\mathcal{L}_\sigma(F) \neq \emptyset$.*

The grounded, ideal and eager semantics are *unique status* semantics, meaning that they map every argumentation framework to a unique labelling.

Proposition 2. *Let F be an argumentation framework and let $\sigma \in \{gr, id, ea\}$. Then $|\mathcal{L}_\sigma(F)| = 1$.*

3. Addition persistence

We say that a semantics σ satisfies XY addition persistence if every σ labelling of an argumentation framework F in which x is labelled X and y is labelled Y is still a σ labelling of F after adding an attack from x to y. Formally:

Definition 8. Let σ be a semantics and let $X, Y \in \{\mathbf{O}, \mathbf{I}, \mathbf{U}\}$. We say that σ satisfies XY *addition persistence* if and only if for all $(A, \rightsquigarrow) \in \mathcal{F}$ and $x, y \in A$, if $L \in \mathcal{L}_\sigma((A, \rightsquigarrow))$, $L(x) = X$ and $L(y) = Y$, then $L \in \mathcal{L}_\sigma((A, \rightsquigarrow \cup \{(x, y)\}))$.

Table 1 shows the full overview of addition persistence under the semantics we consider. A gray cell means that the combination is satisfied and a white cell means that it is violated. The white and gray cells refer to the counterexamples and proofs provided in this section.

We can distinguish two types of addition persistence principles. The first are **UI**, **IU**, and **II** addition persistence. Intuitively, these principles deal with the addition of attacks where the label of the newly attacked argument is no longer justified under the complete semantics. They are violated under the complete semantics and consequently under all the other semantics as well. The stable semantics is the only exception here, and satisfies **UI**, **IU** addition persistence due to the fact that no argument is ever labeled **U** under the stable semantics. The **UI**, **IU**, and **II** addition persistence may be considered as desirable principles for semantics that tolerate conflicts, which we do not consider in this paper [3].

Table 1

Addition persistence for labelling-based semantics

	Addition persistence								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	Prp. 6	Prp. 8	Prp. 6	Prp. 8	Ex. 8	Ex. 10	Prp. 7	Ex. 13	Ex. 14
Complete	Prp. 5	Prp. 5	Prp. 5	Prp. 5	Prp. 5	Ex. 10	Prp. 5	Ex. 13	Ex. 14
Preferred	Prp. 10	Prp. 10	Prp. 11	Ex. 7	Prp. 10	Ex. 10	Prp. 11	Ex. 13	Ex. 14
Stable	Prp. 9	Prp. 12	Prp. 12	Prp. 12	Prp. 9	Prp. 12	Prp. 9	Prp. 12	Ex. 14
Semi-stable	Ex. 1	Ex. 3	Ex. 5	Ex. 7	Ex. 9	Ex. 10	Ex. 11	Ex. 13	Ex. 14
Ideal	Ex. 2	Ex. 4	Ex. 6	Ex. 7	Ex. 8	Ex. 10	Ex. 12	Ex. 13	Ex. 14
Eager	Ex. 1	Ex. 3	Ex. 5	Ex. 7	Ex. 9	Ex. 10	Ex. 11	Ex. 13	Ex. 14

The remaining combinations form the second type. These are **OO**, **OU**, **UO**, **UU**, **OI** and **IO** addition persistence. All these principles are satisfied by the complete semantics. Thus, they cover the complete range of possible additions of attacks that do not invalidate a complete labelling. On this ground, these principles may be regarded as reasonable or desirable principles for all semantics we consider in this paper, since they are all based on complete labellings. The question of whether they are satisfied does not only depend on whether the added attack invalidates a complete labelling, however. It also depends on whether the added attack changes the set of complete labellings as a whole, which in turn may lead to a change in the set of labellings that are grounded, preferred, semi-stable, and so on. Indeed, apart from the complete semantics, the only semantics that satisfies all principles of the second type is the stable semantics. All the other semantics violate one or more principles of the second type. In particular:

- The grounded semantics violates **OI** addition persistence. This arises from the fact that the grounded semantics is **U**-maximising. As demonstrated in Example 8, if x is **O** and y is **I** in the grounded labelling of an argumentation framework then adding an attack from x to y may lead to an additional complete labelling in which both x and y are **U**. This new complete labelling has a strictly larger set of **U**-labelled arguments and thus becomes the new grounded labelling that replaces the old one. Note that this phenomenon does not occur under the preferred semantics, which does satisfy **OI** addition persistence.
- The preferred semantics violates **UU** addition persistence. The **I**-maximising behaviour of the preferred semantics is responsible for this. As demonstrated in Example 7, if x and y are both **U** in a preferred labelling of an argumentation framework then adding an attack from x to y may lead to a new complete labelling in which x is **I**. This new complete labelling has a strictly larger set of arguments that are labelled **I**, and thus becomes a new preferred labelling that replaces the initial one. Note that this phenomenon does not occur under the grounded semantics, which does satisfy **UU** addition persistence.
- The semi-stable, eager and ideal violate *all* addition persistence principles. For the semi-stable semantics, the **U**-minimising behaviour is responsible for this. All the examples involve situations where there is a semi-stable labelling with at least one **U**-labelled argument, while after adding the attack, there is an additional complete labelling with no **U**-labelled arguments, which becomes a new semi-stable labelling that replaces the initial one. The same examples can also be used to show violation of addition persistence principles under the ideal and eager semantics.

In the remainder of this section, we present the counterexamples for the white cells in Table 1 and proofs for the gray cells.

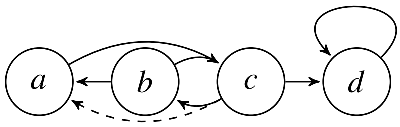

Fig. 1. **OO** addition persistence *ss*, *id* and *ea*.

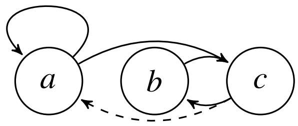

Fig. 2. **OU** addition persistence *ss*, *id* and *ea*.

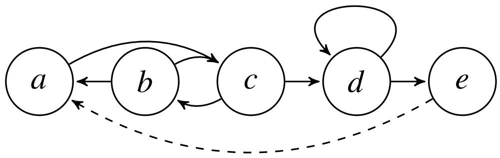

Fig. 3. **UO** addition persistence *ss* and *id*.

Example 1 (Failure of **OO** addition persistence *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 1 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after adding the attack drawn with a dotted line. The new unique semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O})\}$.

Example 2 (Failure of **OO** addition persistence *id* semantics)**.** In the argumentation framework shown in Fig. 1 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer the ideal labelling after adding the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U}), (d, \mathbf{U})\}$.

Example 3 (Failure of **OU** addition persistence *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 2 the labelling $\{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O})\}$ is no longer a semi-stable or eager labelling after adding the attack drawn with a dotted line. The new unique semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{O}), (c, \mathbf{I})\}$.

Example 4 (Failure of **OU** addition persistence *id* semantics)**.** In the argumentation framework shown in Fig. 2 the labelling $\{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O})\}$ is no longer the ideal labelling after adding the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U})\}$.

Example 5 (Failure of **UO** addition persistence *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 3 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U}), (e, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after adding the attack drawn with a dotted line. The new unique semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I})\}$.

Example 6 (Failure of **UO** addition persistence *id* semantics)**.** In the argumentation framework shown in Fig. 3 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U}), (e, \mathbf{U})\}$ is no longer the ideal labelling after adding the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U}), (d, \mathbf{U}), (e, \mathbf{U})\}$.

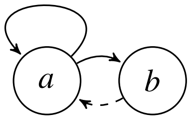

Fig. 4. **UU** addition persistence *pr*, *id*, *ss* and *ea*.

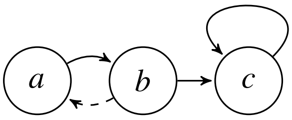

Fig. 5. **OI** addition persistence *gr*, *id*, *ss* and *ea*.

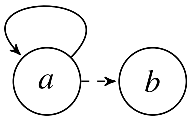

Fig. 6. **UI** addition persistence *gr*, *co*, *pr*, *ss*, *id* and *ea*.

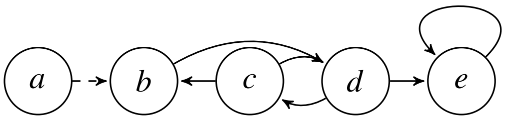

Fig. 7. **IO** addition persistence *ss* and *id*.

Example 7 (Failure of **UU** addition persistence *pr*, *id*, *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 4 the labelling $\{(a, \mathbf{U}), (b, \mathbf{U})\}$ is no longer a *pr*, *id*, *ss* or *ea* labelling after adding the attack drawn with a dotted line. The new *pr*, *id*, *ss* and *ea* labelling is $\{(a, \mathbf{O}), (b, \mathbf{I})\}$.

Example 8 (Failure of **OI** addition persistence *gr*, *id* semantics)**.** In the argumentation framework shown in Fig. 5 the labelling $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{U})\}$ is no longer a *gr* or *id* labelling after adding the attack drawn with a dotted line. The new *gr* and *id* labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U})\}$.

Example 9 (Failure of **OI** addition persistence *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 5 the labelling $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after adding the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O})\}$.

Example 10 (Failure of **UI** addition persistence *gr*, *co*, *pr*, *ss*, *id* and *ea* semantics)**.** In the argumentation framework shown in Fig. 6 the labelling $\{(a, \mathbf{U}), (b, \mathbf{I})\}$ is no longer a *gr*, *co*, *pr*, *ss*, *id* or *ea* labelling after adding the attack drawn with a dotted line. The new labelling under these semantics is $\{(a, \mathbf{U}), (b, \mathbf{U})\}$.

Example 11 (Failure of **IO** addition persistence *ss* and *ea* semantics)**.** In the argumentation framework shown in Fig. 7 the labelling $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after adding the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O})\}$.

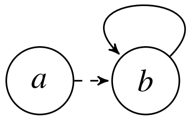

Fig. 8. **IU** addition persistence.

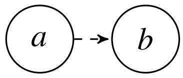

Fig. 9. **II** addition persistence.

Example 12 (Failure of **IO** addition persistence *id* semantics)**.** In the argumentation framework shown in Fig. 7 the labelling $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{U})\}$ is no longer the ideal labelling after adding the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{U}), (d, \mathbf{U}), (e, \mathbf{U})\}$.

Example 13 (Failure of **IU** addition persistence under all semantics except stable)**.** In the argumentation framework shown in Fig. 8 the labelling $\{(a, \mathbf{I}), (b, \mathbf{U})\}$ is no longer a *gr*, *co*, *pr*, *ss* and *id* labelling after adding the attack drawn with a dotted line. The new labelling under these semantics is $\{(a, \mathbf{I}), (b, \mathbf{O})\}$.

Example 14 (Failure of **II** addition persistence all semantics)**.** In the argumentation framework shown in Fig. 9 the labelling $\{(a, \mathbf{I}), (b, \mathbf{I})\}$ is no longer a labelling under any semantics after adding the attack drawn with a dotted line. The new labelling under all semantics is $\{(a, \mathbf{I}), (b, \mathbf{O})\}$.

The following two propositions will be used in the proofs that follow. The first states that, given any pair of complete labellings L, L' of an argumentation framework, it holds that L' accepts more arguments than L if and only if L' rejects more arguments than L.

Proposition 3. *Let* $(A, \rightsquigarrow)$ *be an argumentation framework and let* $L, L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$*. Then* $L^{-1}(\mathbf{I}) \subseteq L'^{-1}(\mathbf{I})$ *if and only if* $L^{-1}(\mathbf{O}) \subseteq L'^{-1}(\mathbf{O})$.

Proof. For the only if direction, assume $L^{-1}(\mathbf{I}) \subseteq L'^{-1}(\mathbf{I})$. Let $x \in A$ and assume $L(x) = \mathbf{O}$. Definition 4 implies that there is a $y \in A$ such that $y \rightsquigarrow x$ and $L(y) = \mathbf{I}$. Our assumption then implies $L'(y) = \mathbf{I}$. Using again Definition 4 it follows that $L'(x) = \mathbf{O}$. Hence $L^{-1}(\mathbf{O}) \subseteq L'^{-1}(\mathbf{O})$.

For the if direction, assume $L^{-1}(\mathbf{O}) \subseteq L'^{-1}(\mathbf{O})$. Let $x \in A$ and assume $L(x) = \mathbf{I}$. Definition 4 implies that for all $y \in A$ such that $y \rightsquigarrow x$ we have $L(y) = \mathbf{O}$. Our assumption then implies that, for all $y \in A$ such that $y \rightsquigarrow x$, $L'(y) = \mathbf{O}$. Using again Definition 4 it follows that $L'(x) = \mathbf{I}$. Hence $L^{-1}(\mathbf{I}) \subseteq L'^{-1}(\mathbf{I})$. □

The second proposition concerns some facts that we can learn about the grounded labelling of an argumentation framework F on the basis of a complete labelling of F.

Proposition 4. *Let* F *be an argumentation framework, let* $L \in \mathcal{L}_{co}(F)$ *and* $L' \in \mathcal{L}_{gr}(F)$*. Then:*

(1) *If* $L(x) = \mathbf{O}$ *then* $L'(x) \neq \mathbf{I}$.
(2) *If* $L(x) = \mathbf{I}$ *then* $L'(x) \neq \mathbf{O}$.
(3) *If* $L(x) = \mathbf{U}$ *then* $L'(x) = \mathbf{U}$.

Proof. Let F be an argumentation framework, let $L \in \mathcal{L}_{co}(F)$ and $L' \in \mathcal{L}_{gr}(F)$.

(1) From the definition of grounded labelling it follows that $L'(x) = \mathbf{I}$ implies $L(x) = \mathbf{I}$. By contraposition it follows that $L(x) = \mathbf{O}$ implies $L'(x) \neq \mathbf{I}$.

(2) From the definition of grounded labelling together with Proposition 3 it follows that $L'(x) = \mathbf{O}$ implies $L(x) = \mathbf{O}$. By contraposition it follows that $L(x) = \mathbf{I}$ implies $L'(x) \neq \mathbf{O}$.

(3) If $L(x) = \mathbf{U}$ then $L(x) \neq \mathbf{I}$ and hence, from the definition of grounded labelling it follows that $L'(x) \neq \mathbf{I}$. Furthermore $L(x) \neq \mathbf{O}$ and hence, from the definition of grounded labelling together with Proposition 3, it follows that $L'(x) \neq \mathbf{O}$. Hence $L'(x) = \mathbf{U}$. □

Proposition 5. *The co semantics satisfies* **OU**, **OI**, **OO**, **UO**, **UU** *and* **IO** *addition persistence.*

Proof. Follows directly from the definition of a complete labelling (Definition 4). □

Proposition 6. *The gr semantics satisfies* **OO** *and* **UO** *addition persistence.*

Proof. Let $F = (A, \rightsquigarrow)$ and $F' = (A, \rightsquigarrow \cup \{(x, y)\})$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Let L be the grounded labelling of F and assume $L(x) \in \{\mathbf{O}, \mathbf{U}\}$ and $L(y) = \mathbf{O}$. Let z be an argument such that $z \neq x$, $L(z) = \mathbf{I}$, and $z \rightsquigarrow y$ (existence of z follows from Definition 4). Because the complete semantics satisfies **OO** and **UO** addition persistence (Proposition 5) L is a complete labelling of $(A, \rightsquigarrow')$. Suppose L is not the grounded labelling of $(A, \rightsquigarrow')$. We show that this leads to a contradiction. Then let L' be the grounded labelling of $(A, \rightsquigarrow')$. Proposition 4 implies that $L'(x) \in \{\mathbf{U}, \mathbf{O}\}$ and $L'(y) \in \{\mathbf{U}, \mathbf{O}\}$. We distinguish two cases:

(1) $L'(y) = \mathbf{U}$: Using Definition 4 it follows that $L'(z) \neq \mathbf{I}$. Furthermore since $L(z) = \mathbf{I}$, Proposition 4 implies $L'(z) = \mathbf{U}$. Using Definition 4 again it follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$.
(2) $L'(y) = \mathbf{O}$: Since $L(x) \in \{\mathbf{O}, \mathbf{U}\}$ it then follows from Definition 4 that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$.

It thus follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L cannot be the grounded labelling of $(A, \rightsquigarrow)$. Hence L is the grounded labelling of $(A, \rightsquigarrow')$. □

Proposition 7. *The gr semantics satisfies* **IO** *addition persistence.*

Proof. Let $F = (A, \rightsquigarrow)$ and $F' = (A, \rightsquigarrow \cup \{(x, y)\})$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Let L be the grounded labelling of F and assume $L(x) = \mathbf{I}$ and $L(y) = \mathbf{O}$. Let z be an argument such that $z \neq x$, $L(z) = \mathbf{I}$, and $z \rightsquigarrow y$ (existence of z follows from Definition 4). Because the complete semantics satisfies **IO** addition persistence (Proposition 5) L is a complete labelling of $(A, \rightsquigarrow')$. Suppose L is not the grounded labelling of $(A, \rightsquigarrow')$. We show that this leads to a contradiction. Let L' be the actual grounded labelling of $(A, \rightsquigarrow')$. Proposition 4 implies that $L'(x) \in \{\mathbf{I}, \mathbf{U}\}$ and $L'(y) \in \{\mathbf{O}, \mathbf{U}\}$. We distinguish four cases:

(1) $L'(x) = \mathbf{U}$ and $L'(y) = \mathbf{U}$: Using Definition 4 it follows that $L'(z) \neq \mathbf{I}$. Furthermore since $L(z) = \mathbf{I}$, Proposition 4 implies $L'(z) = \mathbf{U}$. Definition 4 then implies that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L cannot be the grounded labelling of $(A, \rightsquigarrow)$.
(2) $L'(x) = \mathbf{I}$ and $L'(y) = \mathbf{U}$: This case is not possible as it implies that L' is not a complete labelling of $(A, \rightsquigarrow')$.
(3) $L'(x) = \mathbf{U}$ and $L'(y) = \mathbf{O}$: Definition 4 then implies that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L cannot be the grounded labelling of $(A, \rightsquigarrow)$.

(4) $L'(x) = \mathbf{I}$ and $L'(y) = \mathbf{O}$: We distinguish two sub-cases:

(a) $L'(z) = \mathbf{I}$: Using Definition 4 it then follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$.

(b) $L'(z) = \mathbf{U}$: For this case we transform L' into a new labelling L'' by repeatedly relabelling arguments as follows. For every argument x:

i. if x is $\mathbf{O}$ and x is not attacked (in $(A, \rightsquigarrow)$) by an argument labelled $\mathbf{I}$ then relabel x as $\mathbf{U}$.
ii. if x is $\mathbf{I}$ and x is attacked (in $(A, \rightsquigarrow)$) by an argument labelled $\mathbf{U}$ then relabel x as $\mathbf{U}$.

If we repeat these steps until no more arguments are relabelled then the resulting labelling L'' is a complete labelling of $(A, \rightsquigarrow)$ such that $L''^{-1}(\mathbf{U}) \subset L^{-1}(\mathbf{U})$. But this is a contradiction because it implies that L cannot be the grounded labelling of $(A, \rightsquigarrow)$.

Therefore L is the grounded labelling of $(A, \rightsquigarrow')$. $\square$

Proposition 8. *The gr semantics satisfies* **UU** *and* **OU** *addition persistence.*

Proof. Let $F = (A, \rightsquigarrow)$ and $F' = (A, \rightsquigarrow \cup \{(x, y)\})$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Let L and L' be the grounded labelling of F and F', respectively. We assume $L(x) \in \{\mathbf{O}, \mathbf{U}\}$ and $L(y) = \mathbf{U}$ and prove that $L = L'$. If $x \rightsquigarrow y$ we are done. For the case $x \not\rightsquigarrow y$, we assume $L \neq L'$ and derive a contradiction. Let $z \in A$ be an argument such that $z \neq x$, $L(z) = \mathbf{U}$ and $z \rightsquigarrow y$ (existence of z follows from Definition 4). Because the complete semantics satisfies **UU** and **OU** addition persistence (Proposition 5) we get $L \in \mathcal{L}_{co}(F')$. Using Proposition 4 it then follows that $L'(y) = L'(z) = \mathbf{U}$. Using Definition 4 it then follows that L' is also a complete labelling of F. But this is a contradiction because it implies that L cannot be the grounded labelling of F. Hence $L = L'$. $\square$

Proposition 9. *The stb semantics satisfies* **OO***,* **IO** *and* **OI** *addition persistence.*

Proof. Follows directly from definition of a stable labelling (Definition 7). $\square$

Proposition 10. *The pr semantics satisfies* **OI***,* **OO** *and* **OU** *addition persistence.*

Proof. Let $F = (A, \rightsquigarrow)$ and $F' = (A, \rightsquigarrow \cup \{(x, y)\})$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Suppose $L \in \mathcal{L}_{pr}(F)$. Assume $L(x) = \mathbf{O}$. We prove that $L \in \mathcal{L}_{pr}(F')$. If $x \rightsquigarrow y$ we are done. For the case $x \not\rightsquigarrow y$, assume the contrary. Because the complete semantics satisfies **OI**, **OO** and **OU** addition persistence (Proposition 5) we have $L \in \mathcal{L}_{co}(F')$ and thus there must be an $L' \in \mathcal{L}_{pr}(F')$ such that $L^{-1}(\mathbf{I}) \subset L'^{-1}(\mathbf{I})$. Proposition 3 then implies $L^{-1}(\mathbf{O}) \subset L'^{-1}(\mathbf{O})$ and hence $L'(x) = \mathbf{O}$. Since $L(x) = \mathbf{O}$, Definition 4 then implies that $L' \in \mathcal{L}_{co}(F)$. But this is a contradiction because it implies that L cannot be a preferred labelling of F. Hence $L \in \mathcal{L}_{pr}(F')$. $\square$

Proposition 11. *The pr semantics satisfies* **UO** *and* **IO** *addition persistence.*

Proof. Let $F = (A, \rightsquigarrow)$ and $F' = (A, \rightsquigarrow \cup \{(x, y)\})$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Suppose $L \in \mathcal{L}_{pr}(F)$. Assume $L(y) = \mathbf{O}$. We prove that $L \in \mathcal{L}_{pr}(F')$. If $x \rightsquigarrow y$ we are done. For the case $x \not\rightsquigarrow y$, assume the contrary. Let $z \in A$ be an argument such that $z \neq x$, $L(z) = \mathbf{I}$ and $z \rightsquigarrow y$ (existence of z follows from Definition 4). Because the complete semantics satisfies **UO** and **IO** addition persistence (Proposition 5) we get $L \in \mathcal{L}_{co}(F')$ and thus there must be an $L' \in \mathcal{L}_{pr}(F')$ such that $L^{-1}(\mathbf{I}) \subset L'^{-1}(\mathbf{I})$. This implies $L'(z) = \mathbf{I}$ and therefore $L'(y) = \mathbf{O}$. Because the complete semantics satisfies **IO**, **UO** and **OO** addition persistence (Proposition 5) it follows that L' is a complete labelling of F. But this is a contradiction because it implies that L cannot be a preferred labelling of F. Hence $L \in \mathcal{L}_{pr}(F')$. □

Proposition 12. *The stb semantics satisfies* **OU***,* **UO***,* **UU***,* **UI** *and* **IU** *addition persistence.*

Proof. This follow vacuously because, in a stable labelling, no argument is labelled **U**. □

4. Removal persistence

We say that a semantics σ satisfies XY removal persistence whenever every σ labelling of an argumentation framework F in which two arguments x and y are labelled X and Y, respectively, is still a σ labelling of F after removing the attack from x to y. Formally:

Definition 9. Let σ be a semantics and let $X, Y \in \{\mathbf{O}, \mathbf{I}, \mathbf{U}\}$. We say that σ satisfies XY *removal persistence* if and only if for all $(A, \rightsquigarrow) \in \mathcal{F}$ and $x, y \in A$, if $L \in \mathcal{L}_\sigma((A, \rightsquigarrow))$, $L(x) = X$ and $L(y) = Y$, then $L \in \mathcal{L}_\sigma((A, \rightsquigarrow \setminus \{(x, y)\}))$.

Table 2 shows the full overview of removal persistence under the semantics we consider. Like before, gray cells refer to proofs and indicate that removal persistence holds, while white cells refer to counterexamples and indicate that removal persistence does not hold.

We can distinguish three types of removal persistence principles. First of all, **UI**, **IU** and **II** removal persistence are trivially satisfied under all semantics we consider, where these combinations of labels are never assigned to two arguments where one attacks the other.

The **UU** and **IO** removal persistence principles are of the second type. These principles fail under the complete semantics, where the attack that is removed may be required for the justification of label of second argument. As a direct consequence, these principles fail under all semantics, since they are all based on complete labellings. The only exception here is **UU** removal persistence, which is vacuously satisfied under the stable semantics.

The **OO**, **OU**, **UO** and **OI** removal persistence principles are of the third type. They are all satisfied by the complete semantics. Thus, they cover the complete range of possible removals of attacks that do not invalidate a complete labelling. On this ground, these principles may be regarded as reasonable or desirable principles for all semantics we consider in this paper, since they are all based on complete labellings. Whether they are satisfied under these semantics does not only depend on whether the removed

Table 2

Removal persistence for labelling-based semantics

	Removal persistence								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	Prp. 13	Prp. 13	Prp. 19	Ex. 21	Prp. 18	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Complete	Prp. 14	Prp. 20	Prp. 20	Ex. 21	Prp. 20	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Preferred	Prp. 15	Prp. 15	Prp. 16	Ex. 21	Prp. 15	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Stable	Prp. 17	Prp. 22	Prp. 22	Prp. 22	Prp. 17	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Semi-stable	Ex. 15	Ex. 17	Ex. 19	Ex. 21	Ex. 22	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Ideal	Ex. 16	Ex. 18	Ex. 20	Ex. 21	Ex. 23	Prp. 21	Ex. 24	Prp. 21	Prp. 21
Eager	Ex. 15	Ex. 17	Ex. 19	Ex. 21	Ex. 22	Prp. 21	Ex. 24	Prp. 21	Prp. 21

attack invalidates a complete labelling, however. It also depends on whether the removal changes the set of complete labellings as a whole, which in turn may lead to a change in the set of labellings that are grounded, preferred, semi-stable, and so on. Thus, the situation is similar to that of addition persistence in the previous section. Interestingly, the results here show a sharp contrast between the grounded, preferred and stable semantics, which satisfy all principles of the third type, and the semi-stable, ideal and eager semantics, which satisfy none.

In the remainder of this section, we present the counterexamples for the white cells in Table 2 and proofs for the gray cells.

Example 15 (Failure of **OO** removal persistence under the semi-stable and eager semantics)**.** In the argumentation framework shown in Fig. 10 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after removing the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O})\}$.

Example 16 (Failure of **OO** removal persistence under the ideal semantics)**.** In the argumentation framework shown in Fig. 10 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer the ideal labelling after removing the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U}), (d, \mathbf{U})\}$.

Example 17 (Failure of **OU** removal persistence semi-stable and eager semantics)**.** In the argumentation framework shown in Fig. 11 the labelling $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O}), (f, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after removing the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I}), (f, \mathbf{O})\}$.

Example 18 (Failure of **OU** removal persistence ideal semantics)**.** In the argumentation framework shown in Fig. 11 the labelling $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O}), (f, \mathbf{U})\}$ is no longer the ideal labelling after removing the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U}), (d, \mathbf{U}), (e, \mathbf{U}), (f, \mathbf{U})\}$.

Example 19 (Failure of **UO** removal persistence semi-stable and eager semantics)**.** In the argumentation framework shown in Fig. 12 the labelling $\{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after removing the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{U}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O})\}$.

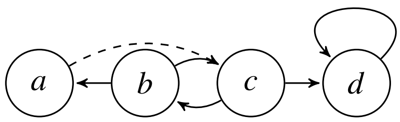

Fig. 10. **OO** removal persistence *ss*, *id* and *ea*.

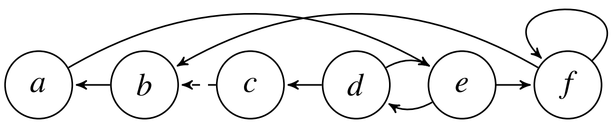

Fig. 11. **OU** removal persistence *ss*, *id* and *ea*.

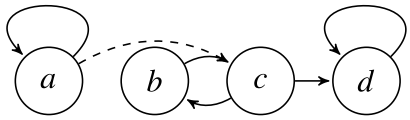

Fig. 12. **UO** removal persistence *ss*, *id* and *ea*.

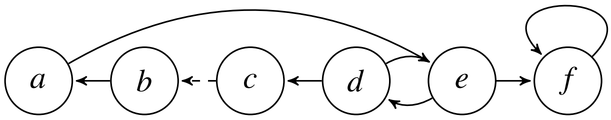

Fig. 13. **OI** removal persistence *ss*, *id* and *ea*.

Example 20 (Failure of **UO** removal persistence ideal semantics)**.** In the argumentation framework shown in Fig. 12 the labelling $\{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is no longer the ideal labelling after removing the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{U}), (d, \mathbf{U})\}$.

Example 21 (Failure of **UU** removal persistence under all but the stable semantics)**.** This case is trivial: consider an argumentation framework consisting of one self-attacking argument. This argument is undecided, but is no longer undecided if the self-attack is removed.

Example 22 (Failure of **OI** removal persistence semi-stable and eager semantics)**.** In the argumentation framework shown in Fig. 13 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O}), (f, \mathbf{U})\}$ is no longer a semi-stable or eager labelling after removing the attack drawn with a dotted line. The new semi-stable and eager labelling is $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I}), (f, \mathbf{O})\}$.

Example 23 (Failure of **OI** removal persistence ideal semantics)**.** In the argumentation framework shown in Fig. 13 the labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O}), (f, \mathbf{U})\}$ is no longer the ideal labelling after removing the attack drawn with a dotted line. The new ideal labelling is $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{U}), (d, \mathbf{U}), (e, \mathbf{U}), (f, \mathbf{U})\}$.

Example 24 (Failure of **IO** removal persistence under all semantics)**.** This case is trivial: consider an argumentation framework consisting of two arguments a and b where a attacks b. Under all semantics considered here, a is labelled **I** and b is labelled **O**. If we remove the attack from a to b then a and b are both labelled **I**

Proposition 13. *The gr semantics satisfies* **OO** *and* **OU** *removal persistence.*

Proof. Let L be the grounded labelling of $(A, \rightsquigarrow)$, let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$, and let L' be the grounded labelling of $(A, \rightsquigarrow')$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). We assume $L(x) = \mathbf{O}$ and $L(y) \in \{\mathbf{O}, \mathbf{U}\}$ and prove that $L = L'$. If $x \not\rightsquigarrow y$ we are done. For the case $x \rightsquigarrow y$, we assume $L \neq L'$ and derive a contradiction. Since $L(x) = \mathbf{O}$, Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. Proposition 4 then implies that $L'(x) \in \{\mathbf{O}, \mathbf{U}\}$ and $L'(y) \in \{\mathbf{O}, \mathbf{U}\}$. Because the complete semantics satisfies **OO**, **OU**, **UO** and **UU** addition persistence (Proposition 5) we have $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L is not the grounded labelling of $(A, \rightsquigarrow)$. Hence $L = L'$. $\square$

Proposition 14. *The co semantics satisfies* **OO** *removal persistence.*

Proof. Follows directly from Definition 4. □

Proposition 15. *The pr semantics satisfies* **OO**, **OU** *and* **OI** *removal persistence.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework and let $L \in \mathcal{L}_{pr}((A, \rightsquigarrow))$. Assume $L(x) = \mathbf{O}$. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). We prove that $L \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$. If $x \not\rightsquigarrow y$ we are done. For the case $x \rightsquigarrow y$, assume the contrary. Since $L(x) = \mathbf{O}$, Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. It follows that there is an $L' \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$ such that $L^{-1}(\mathbf{I}) \subset L'^{-1}(\mathbf{I})$. Proposition 3 then implies $L^{-1}(\mathbf{O}) \subset L'^{-1}(\mathbf{O})$ and hence $L'(x) = \mathbf{O}$. Because the complete semantics satisfies **OO**, **OU** and **OI** addition persistence (Proposition 5) it follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L is not a preferred labelling of $(A, \rightsquigarrow)$. Hence $L \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$. □

Proposition 16. *The pr semantics satisfies* **UO** *removal persistence.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework and let $L \in \mathcal{L}_{pr}((A, \rightsquigarrow))$. Assume $L(x) = \mathbf{U}$ and $L(y) = \mathbf{O}$. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). We prove that $L \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$. If $x \not\rightsquigarrow y$ we are done. For the case $x \rightsquigarrow y$, assume the contrary. Since $L(y) = \mathbf{O}$, Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. It follows that there is an $L' \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$ such that $L^{-1}(\mathbf{I}) \subset L'^{-1}(\mathbf{I})$. Proposition 3 then implies $L^{-1}(\mathbf{O}) \subset L'^{-1}(\mathbf{O})$ and hence $L'(y) = \mathbf{O}$. Because the complete semantics satisfies **OO**, **UO** and **IO** addition persistence (Proposition 5) it follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction because it implies that L is not a preferred labelling of $(A, \rightsquigarrow)$. Hence $L \in \mathcal{L}_{pr}((A, \rightsquigarrow'))$. □

Proposition 17. *The stb semantics satisfies* **OO** *and* **OI** *removal persistence.*

Proof. Follows directly from Definition 7. □

Proposition 18. *The gr semantics satisfies* **OI** *removal persistence.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). Let L be the grounded labelling of $(A, \rightsquigarrow)$ and assume $L(x) = \mathbf{O}$ and $L(y) = \mathbf{I}$. Since $L(x) = \mathbf{O}$, Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. We now suppose L is not the grounded labelling of $(A, \rightsquigarrow')$ and prove that this leads to a contradiction. Then let L' be the actual grounded labelling of $(A, \rightsquigarrow')$. We then have $L'(x) \in \{\mathbf{O}, \mathbf{U}\}$ and $L'(y) \in \{\mathbf{I}, \mathbf{U}\}$. We distinguish two cases:

(1) $L'(x) = \mathbf{O}$ or $L'(y) = \mathbf{U}$: Because the complete semantics satisfies **OI**, **OU** and **UU** addition persistence (Proposition 5) it follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction, since L is the grounded labelling of $(A, \rightsquigarrow)$ and we have $L'^{-1}(\mathbf{U}) \subset L^{-1}(\mathbf{U})$.

(2) $L'(x) = \mathbf{U}$ and $L'(y) = \mathbf{I}$: For this case we transform L' into a new labelling L'' by repeatedly relabelling arguments as follows. For every argument x:

 (a) if x is **I** and x is attacked (in $(A, \rightsquigarrow)$) by an argument labelled **U** then relabel x as **U**.
 (b) if x is **O** and x is not attacked (in $(A, \rightsquigarrow)$) by an argument labelled **I** then relabel x as **U**.

If we repeat these steps until no more arguments are relabelled then the resulting labelling L'' is a complete labelling of $(A, \rightsquigarrow)$ such that $L''(x) = L''(y) = \mathbf{U}$. But this is a contradiction, since L is the grounded labelling of $(A, \rightsquigarrow)$ and we have $L''^{-1}(\mathbf{U}) \subset L^{-1}(\mathbf{U})$.

Since both cases lead to contradiction it follows that L is the grounded labelling of $(A, \rightsquigarrow')$. □

Proposition 19. *The gr semantics satisfies* **UO** *removal persistence.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). Let L be the grounded labelling of $(A, \rightsquigarrow)$ and assume $L(x) = \mathbf{U}$ and $L(y) = \mathbf{O}$. Since $L(y) = \mathbf{O}$, Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. We now suppose L is not the grounded labelling of $(A, \rightsquigarrow')$ and prove that this leads to a contradiction. Then let L' be the actual grounded labelling of $(A, \rightsquigarrow')$. We then have $L'(x) = \mathbf{U}$ and $L'(y) \in \{\mathbf{O}, \mathbf{U}\}$. Because the complete semantics satisfies **UO** and **UU** addition persistence (Proposition 5 and 5) it follows that $L' \in \mathcal{L}_{co}((A, \rightsquigarrow))$. But this is a contradiction, since L is the grounded labelling of $(A, \rightsquigarrow)$ and we have $L'^{-1}(\mathbf{U}) \subset L^{-1}(\mathbf{U})$. Hence L is the grounded labelling of $(A, \rightsquigarrow')$. □

Proposition 20. *The co semantics satisfies* **OU**, **OI** *and* **UO** *removal persistence.*

Proof. Follows directly from Definition 4. □

Proposition 21. *Any complete-based semantics satisfies* **UI**, **IU**, *and* **II** *removal persistence.*

Proof. This holds vacuously because, in a complete labelling, these combinations of labels are never assigned to two arguments of which one attacks the other. □

Proposition 22. *The stb semantics satisfies* **OU**, **UO** *and* **UU** *removal persistence.*

Proof. This holds vacuously because, in a stable labelling, no argument is labelled **U**. □

5. Addition monotonicity

We now consider addition monotonicity for labelling-based semantics. Suppose x and y are labelled X and Y, respectively, in every σ labelling of an argumentation framework F. The XY addition persistence principle then implies that no σ labelling gets destroyed if we add an attack from x to y to F. The principles we consider now state, on the other hand, that no additional σ labelling is *created*, if we add an attack from x to y.

Definition 10. Let σ be a semantics and let $X, Y \in \{\mathbf{O}, \mathbf{I}, \mathbf{U}\}$. We say that σ satisfies XY *addition monotonicity* if and only if for all $(A, \rightsquigarrow) \in \mathcal{F}$ such that $\mathcal{L}_\sigma((A, \rightsquigarrow)) \neq \emptyset$, and for all $x, y \in A$:

$$\text{If for all } L \in \mathcal{L}_\sigma\big((A, \rightsquigarrow)\big), L(x) = X \text{ and } L(y) = Y \text{ then } \mathcal{L}_\sigma\big(\big(A, \rightsquigarrow \cup \big\{(x, y)\big\}\big)\big) \subseteq \mathcal{L}_\sigma\big((A, \rightsquigarrow)\big).$$

Remark 1. Addition monotonicity is concerned only with how the set of σ labellings of an argumentation framework $(A, \rightsquigarrow)$ changes if the initial set of labellings is nonempty (i.e., if $\mathcal{L}_\sigma((A, \rightsquigarrow)) \neq \emptyset$). This allows us to obtain informative results for the stable semantics, which would otherwise vacuously fail all variants of addition monotonicity. To see why, consider the argumentation framework $(\{a, b\}, \{(b, b)\})$. We have $\mathcal{L}_{stb}((\{a, b\}, \{(b, b)\})) = \emptyset$ and hence

$$\text{for all } L \in \mathcal{L}_{stb}\big(\big(\{a, b\}, \big\{(b, b)\big\}\big)\big), L(a) = \mathbf{O} \text{ and } L(b) = \mathbf{O}. \tag{1}$$

Moreover we have

$$\mathcal{L}_{stb}\big(\big(\{a, b\}, \big\{(b, b)\big\} \cup \big\{(a, b)\big\}\big)\big) = \big\{\big\{(a, \mathbf{I}), (b, \mathbf{O})\big\}\big\} \nsubseteq \mathcal{L}_{stb}\big(\big(\{a, b\}, \big\{(b, b)\big\}\big)\big). \tag{2}$$

If we do not require the initial set of labellings to be nonempty then this example would imply failure of **OO** addition monotonicity under the stable semantics. This is, however, not informative, as it is due to the fact that (1) holds vacuously.

For any unique status semantics (cf. Proposition 2) it holds that addition monotonicity coincides with addition persistence.

Proposition 23. *If σ is a unique status semantics then σ satisfies XY addition monotonicity if and only if σ satisfies XY addition persistence.*

We now show which of the labelling-based semantics that we consider satisfy or violate addition monotonicity. Table 3 shows the full overview: a gray cell means that the principle is satisfied and a white cell means that it is violated. The white and gray cells refer to the counterexamples and proofs provided in this section.

Three types of addition monotonicity principles can be distinguished. The **UI**, **IU** and **II** addition monotonicity principles represent the first type. Like the corresponding addition persistence principles, they are violated under the complete semantics because they deal with the addition of an attack after which the label of the attacked argument is no longer justified under the complete semantics. A direct consequence is that these principles are violated under all the semantics we consider, since they are all based on complete labellings. The exception is the stable semantics, which vacuously satisfies **UI** and **IU** addition monotonicity.

The remaining principles form the second type: **UU**, **OI**, **OO**, **OU**, **UO** and **IO** addition monotonicity. First note that the addition persistence principles for these combinations of labels are all satisfied under the complete semantics. The addition monotonicity principles are only partially satisfied, however. In particular, **UU** and **OI** addition monotonicity fail under the complete semantics, as demonstrated in Examples 31 and 32. These principles are also violated by all the other semantics, with the exception of **UU**

Table 3

Addition monotonicity for labelling-based semantics

	Addition monotonicity								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	Prp. 24	Prp. 24	Prp. 24	Prp. 24	Ex. 32	Ex. 25	Prp. 24	Ex. 26	Ex. 27
Complete	Prp. 25	Prp. 25	Prp. 26	Ex. 31	Ex. 32	Ex. 25	Prp. 26	Ex. 26	Ex. 27
Preferred	Ex. 28	Ex. 29	Ex. 30	Ex. 31	Ex. 32	Ex. 25	Ex. 33	Ex. 26	Ex. 27
Stable	Ex. 28	Prp. 27	Prp. 27	Prp. 27	Ex. 32	Prp. 27	Ex. 33	Prp. 27	Ex. 27
Semi-stable	Ex. 28	Ex. 29	Ex. 30	Ex. 31	Ex. 32	Ex. 25	Ex. 33	Ex. 26	Ex. 27
Ideal	Ex. 2	Ex. 4	Ex. 6	Ex. 7	Ex. 32	Ex. 25	Ex. 12	Ex. 26	Ex. 27
Eager	Ex. 1	Ex. 3	Ex. 5	Ex. 7	Ex. 9	Ex. 25	Ex. 11	Ex. 26	Ex. 27

addition monotonicity, which is satisfied by the grounded semantics and vacuously satisfied by the stable semantics. The remaining principles, i.e., **OO**, **OU**, **UO** and **IO** addition monotonicity, are all satisfied under the complete semantics. They are also satisfied under the grounded semantics, but this follows from the corresponding results for addition persistence (see Proposition 23). However, the preferred, semi-stable, ideal and eager semantics violate these four principles. The stable semantics furthermore vacuously satisfies **OU**, **UO** addition monotonicity but violates **OO** and **IO** addition monotonicity.

In the remainder of this section, we present the counterexamples for the white cells in Table 3 and proofs for the gray cells.

Example 25 (Failure of **UI** addition monotonicity under the all but the stable semantics.)**.** Consider an argumentation framework with two arguments a, b, where a is self-attacking. The unique labelling under all but the stable semantics assigns **U** to a and **I** to b. If we add an attack from a to b the new unique labelling under all but the stable semantics assigns **U** to both a and b.

Example 26 (Failure of **IU** addition monotonicity under all but the stable semantics.)**.** Consider an argumentation framework with two arguments a, b, where b is self-attacking. The unique labelling under all but the stable semantics assigns **I** to a and **U** to b. If we add an attack from a to b the new unique labelling under all but the stable semantics assigns **I** to a and **O** to b.

Example 27 (Failure of **II** addition monotonicity under all semantics)**.** Consider an argumentation framework with two arguments a, b and no attacks. The unique labelling under all semantics assigns **I** to both a and b. If we add an attack from a to b the new unique labelling under all semantics assigns **I** to a and **O** to b.

Example 28 (Failure of **OO** addition monotonicity under the *pr*, *stb* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 14 (not including the attack drawn with a dotted line). The labelling $L = \{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O})\}$ is the unique preferred, stable and semi-stable labelling of F. If we add an attack from e to c (drawn with a dotted line), which are both labelled **O** in L, then a new preferred, stable and semi-stable labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{O}), (e, \mathbf{I})\}$ appears.

Example 29 (Failure of **OU** addition monotonicity under the *pr* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 15 (not including the attack drawn with a dotted line). The labelling $L = \{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U})\}$ is the unique preferred and semi-stable labelling of F. If we add an

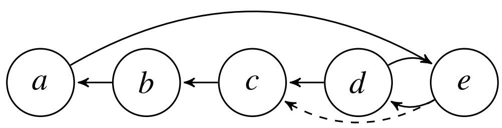

Fig. 14. Failure of **OO** addition monotonicity.

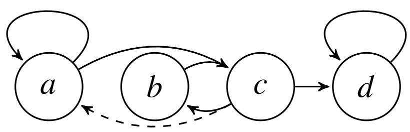

Fig. 15. Failure of **OU** addition monotonicity under the preferred and semi-stable semantics.

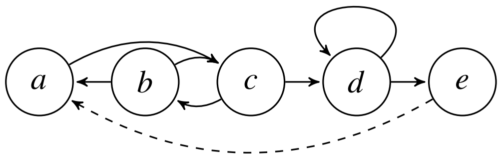

Fig. 16. Failure of **UO** addition monotonicity under the preferred and semi-stable semantics.

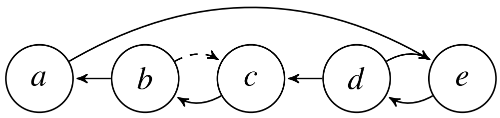

Fig. 17. Failure of **IO** addition monotonicity.

attack attack from c to a (drawn with a dotted line), which are labelled, respectively, **O** and **U** in L, then a new preferred and semi-stable labelling $\{(a, \mathbf{O}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O})\}$ appears.

Example 30 (Failure of **UO** addition monotonicity under the preferred and semi-stable semantics)**.** Let F be the argumentation framework shown in Fig. 16 (not including the attack drawn with a dotted line). The labelling $L = \{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{U}), (e, \mathbf{U})\}$ is the unique preferred and semi-stable labelling of F. If we add an attack attack from e to a (drawn with a dotted line), which are labelled, respectively, **U** and **O** in L, then a new preferred and semi-stable labelling $\{(a, \mathbf{O}), (b, \mathbf{O}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I})\}$ appears.

Example 31 (Failure of **UU** addition monotonicity under the *co*, *pr* and *ss* semantics)**.** Consider an argumentation framework F with two arguments a and b where a attacks b and a is self-attacking. Let L be the labelling $\{(a, \mathbf{U}), (b, \mathbf{U})\}$. We have that L is the unique complete labelling of F as well as the unique preferred and semi-stable labelling. If we add an attack from b to a, a new complete labelling $L' = \{(a, \mathbf{O}), (b, \mathbf{I})\}$ appears, and L' is also preferred and semi-stable.

Example 32 (Failure of **OI** addition monotonicity under all semantics)**.** Consider an argumentation framework F with two arguments a and b where a attacks b. Let F' be the same argumentation framework with an additional attack from b to a. Under all semantics we consider, the unique labelling of F is $\{(a, \mathbf{I}), (b, \mathbf{O})\}$, while F' possesses a labelling (such as $\{(a, \mathbf{U}), (b, \mathbf{U})\}$ or $\{(a, \mathbf{O}), (b, \mathbf{I})\}$) that is not a labelling of F under that semantics.

Example 33 (Failure of **IO** addition monotonicity under the preferred, stable and semi-stable semantics)**.** Let F be the argumentation framework shown in Fig. 17 (not including the attack drawn with a dotted line). The labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O})\}$ is the unique preferred, stable and semi-stable labelling of F. If we add an attack from b to c (drawn with a dotted line), which are labelled, respectively, **I** and **O** in L, then a new preferred, stable and semi-stable labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{O}), (e, \mathbf{I})\}$ appears.

Proposition 24. *The gr semantics satisfies* **OO**, **OU**, **UO**, **UU** *and* **IO** *addition monotonicity.*

Proof. Follows from Proposition 6 and 8 together with Proposition 23. □

Proposition 25. *The co semantics satisfies* **OO** *and* **OU** *addition monotonicity.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \cup \{(x, y)\}$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Suppose for all $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$ we have $L(x) = \mathbf{O}$ and $L(y) \in \{\mathbf{O}, \mathbf{U}\}$. Let $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. We prove that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. Let L' be the grounded labelling of $(A, \rightsquigarrow)$. Our assumption implies $L'(x) = \mathbf{O}$ and $L'(y) \in \{\mathbf{O}, \mathbf{U}\}$. Because the grounded semantics satisfies **OO** and **OU** addition persistence (Proposition 6 and 8) it follows that L' is the grounded labelling of $(A, \rightsquigarrow')$ and hence $L(x) = \mathbf{O}$. Because the complete semantics satisfies **OO**, **OU** and **OI** removal persistence (Proposition 14 and 20), it finally follows that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. □

Proposition 26. *The co semantics satisfies* **IO** *and* **UO** *addition monotonicity.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \cup \{(x, y)\}$. We assume that $(x, y) \notin \rightsquigarrow$ (the case $(x, y) \in \rightsquigarrow$ is trivial). Suppose for all $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$ we have $L(x) \in \{\mathbf{I}, \mathbf{U}\}$ and $L(y) = \mathbf{O}$. Let $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. We prove that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. Let L' be the grounded labelling of $(A, \rightsquigarrow)$. Our assumption implies that $L'(x) \in \{\mathbf{I}, \mathbf{U}\}$ and $L'(y) = \mathbf{O}$. Then let z be argument such that $z \neq x$, $L'(z) = \mathbf{I}$ and $z \rightsquigarrow y$ (existence of z follows from Definition 4). Because the grounded semantics satisfies **IO** and **UO** addition persistence (Proposition 7) it follows that L' is the grounded labelling of $(A, \rightsquigarrow')$. Hence $L(y) = \mathbf{O}$ and $L(z) = \mathbf{I}$. Definition 4 then implies that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. □

Proposition 27. *The stb semantics satisfies* **OU**, **UO**, **UU**, **UI** *and* **IU** *addition monotonicity.*

Proof. This holds vacuously because, in a stable labelling, no argument is labelled **U**. □

6. Removal monotonicity

We finally consider removal monotonicity in the labelling-based setting. Like before, we require that the initial set of labellings is non-empty (see Remark 1).

Definition 11. Let σ be a semantics and let $X, Y \in \{\mathbf{O}, \mathbf{I}, \mathbf{U}\}$. We say that σ satisfies XY *removal monotonicity* if and only if for all $(A, \rightsquigarrow) \in \mathcal{F}$ such that $\mathcal{L}_\sigma((A, \rightsquigarrow)) \neq \emptyset$, and for all $x, y \in A$:

$$\text{If for all } L \in \mathcal{L}_\sigma\big((A, \rightsquigarrow)\big), L(x) = X \text{ and } L(y) = Y \text{ then } \mathcal{L}_\sigma\big((A, \rightsquigarrow \setminus \big\{(x, y)\big\})\big) \subseteq \mathcal{L}_\sigma\big((A, \rightsquigarrow)\big).$$

Removal monotonicity coincides with removal persistence for any unique status semantics (cf. Proposition 2).

Proposition 28. *If σ is a unique status semantics then σ satisfies XY removal monotonicity if and only if σ satisfies XY removal persistence.*

We now show which of the labelling-based semantics that we consider satisfy or violate removal monotonicity. Table 4 shows the full overview: a gray cell means that the principle is satisfied and a white cell means that it is violated. The white and gray cells refer to the counterexamples and proofs provided in this section.

We can distinguish three types of removal monotonicity principles. The **UI**, **IU** and **II** removal monotonicity principles represent the first type. Like removal persistence for these combinations of labels,

Table 4
Removal monotonicity for labelling-based semantics

	Removal monotonicity								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	Prp. 29	Prp. 29	Prp. 29	Ex. 34	Prp. 29	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Complete	Prp. 30	Prp. 30	Prp. 31	Ex. 34	Prp. 30	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Preferred	Ex. 36	Ex. 37	Ex. 38	Ex. 34	Ex. 39	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Stable	Ex. 36	Prp. 33	Prp. 33	Prp. 33	Ex. 39	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Semi-stable	Ex. 36	Ex. 37	Ex. 38	Ex. 34	Ex. 39	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Ideal	Ex. 16	Ex. 18	Ex. 20	Ex. 34	Ex. 23	Prp. 32	Ex. 35	Prp. 32	Prp. 32
Eager	Ex. 15	Ex. 17	Ex. 19	Ex. 34	Ex. 22	Prp. 32	Ex. 35	Prp. 32	Prp. 32

these principles are trivially satisfied by all semantics we consider, under which these combinations of labels are never assigned to two arguments where the first attacks the second.

The **UU** and **IO** removal monotonicity principles represent the second type. Like the removal persistence principles for these combinations of labels, these principles fail under the complete semantics, where the attack that is removed may be required for the justification of label of second argument. Removal of these attacks may therefore lead to new labellings, as demonstrated in Examples 35 and 34. As a consequence **UU** and **IO** removal monotonicity fail under all the other semantics as well, since they are all based on complete labellings. The only exception is the stable semantics, which trivially satisfies **UU** removal monotonicity.

The remaining removal monotonicity principles represent the third type. These are **OO**, **OU**, **UO** and **OI** removal monotonicity. Like the removal persistence principles for these combinations of labels, these principles are all satisfied under the complete semantics. The grounded semantics also satisfies all these principles. None of the other semantics satisfy any of these principles, however, except for the stable semantics, which trivially satisfies **OU** and **UO** removal monotonicity.

In the remainder of this section, we present the counterexamples for the white cells in Table 4 and proofs for the gray cells.

Example 34 (Failure of **UU** removal monotonicity under all but the stable semantics.)**.** Consider an argumentation framework with two arguments a, b, where a is self-attacking and a attacks b. The unique labelling under all but the stable semantics assigns **U** to both a and b. If we remove the attack from a to b the new unique labelling under all but the stable semantics assigns **U** to a and **I** to b.

Example 35 (Failure of **IO** removal monotonicity under all semantics.)**.** Consider an argumentation framework with two arguments a, b, where a attacks b. The unique labelling under all semantics assigns **I** to a and **O** to b. If we remove the attack from a to b the new unique labelling under semantics assigns **I** to both a and b.

Example 36 (Failure of **OO** removal monotonicity under the *pr*, *stb* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 18 (including the attack drawn with a dotted line). The labelling $L = \{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O})\}$ is the unique preferred, semi-stable and stable labelling of F. If we remove the attack from a to c (drawn with a dotted line), which are both labeled **O** in L, then a new preferred, semi-stable and stable labelling $\{(a, \mathbf{I}), (b, \mathbf{O}), (c, \mathbf{I})\}$ appears.

Example 37 (Failure of **OU** removal monotonicity under the *pr* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 19 (including the attack drawn with a dotted line). The labelling

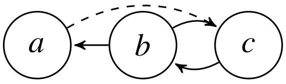

Fig. 18. Failure of **OO** removal monotonicity.

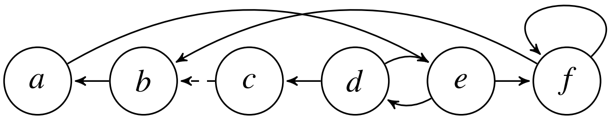

Fig. 19. Failure of **OU** removal monotonicity.

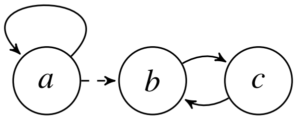

Fig. 20. Failure of **UO** removal monotonicity.

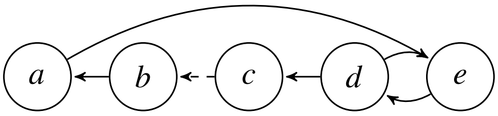

Fig. 21. Failure of **OI** removal monotonicity.

$L = \{(a, \mathbf{U}), (b, \mathbf{U}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O}), (f, \mathbf{U})\}$ is the unique preferred and semi-stable labelling of F. If we remove the attack from c to b (drawn with a dotted line), which are labeled **O** and **U**, respectively, then a new preferred and semi-stable labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I}), (f, \mathbf{O})\}$ appears.

Example 38 (Failure of **UO** removal monotonicity under the *pr* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 20 (including the attack drawn with a dotted line). The labelling $L = \{(a, \mathbf{U}), (b, \mathbf{O}), (c, \mathbf{I})\}$ is the unique preferred and semi-stable labelling of F. If we remove the attack from a to b (drawn with a dotted line), which are labeled, respectively, **U** and **O** in L, then a new preferred and semi-stable labelling $\{(a, \mathbf{U}), (b, \mathbf{I}), (c, \mathbf{O})\}$ appears.

Example 39 (Failure of **OI** removal monotonicity under the *pr*, *stb* and *ss* semantics)**.** Let F be the argumentation framework shown in Fig. 21. has a unique preferred, semi-stable and stable labelling $L = \{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{O}), (d, \mathbf{I}), (e, \mathbf{O})\}$. Removal of the attack from c to b (labelled **O** and **I** in L) leads to an additional preferred, semi-stable and stable labelling $\{(a, \mathbf{O}), (b, \mathbf{I}), (c, \mathbf{I}), (d, \mathbf{O}), (e, \mathbf{I})\}$.

Proposition 29. *The gr semantics satisfies* **OO**, **OU**, **OI** *and* **UO** *removal monotonicity.*

Proof. Follows from Proposition 28 together with Propositions 13, 18 and 19. □

Proposition 30. *The co semantics satisfies* **OO**, **OU** *and* **OI** *removal monotonicity.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). Suppose for all $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$ we have $L(x) = \mathbf{O}$. Let $L \in$

$\mathcal{L}_{co}((A, \rightsquigarrow'))$. We prove that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. Let L' be the grounded labelling of $(A, \rightsquigarrow)$. Our assumption implies that $L'(x) = \mathbf{O}$. Because the grounded semantics satisfies **OO**, **OU** and **OI** removal persistence (Proposition 13 and 18) it follows that L' is the grounded labelling of $(A, \rightsquigarrow')$. Hence $L(x) = \mathbf{O}$. Because the complete semantics satisfies **OO**, **OU** and **OI** addition persistence (Proposition 5), it follows that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. $\square$

Proposition 31. *The co semantics satisfies* **UO** *removal monotonicity.*

Proof. Let $(A, \rightsquigarrow)$ be an argumentation framework. Let $\rightsquigarrow' = \rightsquigarrow \setminus \{(x, y)\}$. We assume that $(x, y) \in \rightsquigarrow$ (the case $(x, y) \notin \rightsquigarrow$ is trivial). Suppose for all $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$ we have $L(x) = \mathbf{U}$ and $L(y) = \mathbf{O}$. Let $L \in \mathcal{L}_{co}((A, \rightsquigarrow'))$. We prove that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. Let L' be the grounded labelling of $(A, \rightsquigarrow)$. Our assumption implies that $L'(x) = \mathbf{U}$ and $L'(y) = \mathbf{O}$. Because the grounded semantics satisfies **UO** removal persistence (Proposition 19) it follows that L' is the grounded labelling of $(A, \rightsquigarrow')$. Hence $L(y) = \mathbf{O}$. Because the complete semantics satisfies **OO**, **UO** and **IO** addition persistence (Proposition 5), it follows that $L \in \mathcal{L}_{co}((A, \rightsquigarrow))$. $\square$

Proposition 32. *Any complete-based semantics satisfies* **UI**, **IU**, *and* **II** *removal monotonicity.*

Proof. This holds vacuously because, in a complete labelling, these combinations of labels are never assigned to two arguments of which one attacks the other. $\square$

Proposition 33. *The stb semantics satisfies* **OU**, **UO** *and* **UU** *removal monotonicity.*

Proof. This holds vacuously because, in a stable labelling, no argument is labelled **U**. $\square$

7. Persistence and monotonicity in the extension-based setting

Most of the semantics we consider in this paper were originally defined in terms of extensions. These extension-based definitions can also be derived from the labelling-based Definitions [31]. Given a semantics σ, a σ-extension of an argumentation framework $(A, \rightsquigarrow)$ is a set E such that $E = \{x \in A \mid L(x) = \mathbf{I}\}$ for some $L \in \mathcal{L}_{\sigma}((A, \rightsquigarrow))$. In this section, we examine persistence and monotonicity principles from the extension-based perspective. We do this by only distinguishing arguments that are in an extension from those that are not. In other words, we distinguish arguments that are labelled **I** from those that are not labelled **I**, but we do not distinguish whether an argument is labelled **O** or **U**. Given a labelling L and argument x we use (with abuse of notation) $L(x) = \not{\mathbf{I}}$ to denote $L(x) \neq \mathbf{I}$. Combining this with the definitions of addition and removal persistence and monotonicity (Definitions 8, 9, 10 and 11) leads to a number additional variants of these principles, namely $\not{\mathbf{I}}\mathbf{I}$, $\mathbf{I}\not{\mathbf{I}}$ and $\not{\mathbf{I}}\not{\mathbf{I}}$ addition/removal persistence and monotonicity. These variants can be understood as being about persistence and monotonicity of extensions. For example, $\not{\mathbf{I}}\mathbf{I}$ addition persistence is equivalent to the property that a σ extension E is still a σ extension after adding an attack from x to y, if $x \notin E$ and $y \in E$. In most cases we can derive results for these additional principles from the results for the three-valued label combinations. In particular, for addition and removal persistence we have the following fact, which follows directly from the definition of addition and removal persistence and therefore holds not just for the semantics we consider, but for every three-valued labelling-based semantics.

Proposition 34. *For every semantics σ we have:*

a) *σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition/removal persistence if and only if σ satisfies* **OO**, **UO**, **OU** *and* **UU** *addition/removal persistence.*
b) *σ satisfies $\not\mathbf{I}\mathbf{I}$ addition/removal persistence if and only if σ satisfies* **OI** *and* **UI** *addition/removal persistence.*
c) *σ satisfies $\mathbf{I}\not\mathbf{I}$ addition/removal persistence if and only if σ satisfies* **IO** *and* **IU** *addition/removal persistence.*

Proof. We prove this for item *a* and for addition persistence. Removal persistence, as well as items *b* and *c*, follow similarly.

For the only if direction, assume that σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition persistence. We will prove that σ satisfies **OO** addition persistence. Let $F = (A, \rightsquigarrow)$, let $x, y \in A$, let $F' = (A, \rightsquigarrow \cup \{(x, y)\})$ and let $L \in \mathcal{L}_\sigma(F)$ be a labelling such that $L(x) = L(y) = \mathbf{O}$. Then $L(x) \neq \mathbf{I}$ and $L(y) \neq \mathbf{I}$. Since σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition persistence it follows that $L \in \mathcal{L}_\sigma(F')$, which proves that σ satisfies **OO** addition persistence. The **UO**, **OU** and **UU** addition persistence principles follow similarly.

For the if direction, assume that σ satisfies **OO**, **UO**, **OU** and **UU** addition persistence. We will prove that σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition persistence. Let $F = (A, \rightsquigarrow)$, let $x, y \in A$, let $F' = (A, \rightsquigarrow \cup \{(x, y)\})$ and let $L \in \mathcal{L}_\sigma(F)$ be a labelling such that $L(x) \neq \mathbf{I}$ and $L(y) \neq \mathbf{I}$. Then $L(x) \in \{\mathbf{O}, \mathbf{U}\}$ and $L(y) = \{\mathbf{O}, \mathbf{U}\}$. Therefore it follows that $L \in \mathcal{L}_\sigma(F')$, which proves that σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition persistence. □

For addition and removal monotonicity things look different. Again, the following fact follows directly from the definition of addition and removal monotonicity.

Proposition 35. *For every semantics σ we have:*

a) *σ satisfies $\not\mathbf{I}\not\mathbf{I}$ addition/removal monotonicity* only if *σ satisfies* **OO**, **UO**, **OU** *and* **UU** *addition/removal monotonicity.*
b) *σ satisfies $\not\mathbf{I}\mathbf{I}$ addition/removal monotonicity* only if *σ satisfies* **OI** *and* **UI** *addition/removal monotonicity.*
c) *σ satisfies $\mathbf{I}\not\mathbf{I}$ addition/removal monotonicity* only if *σ satisfies* **IO** *and* **IU** *addition/removal monotonicity.*

Proof. Similar to the proof of the only if direction in Proposition 34. □

The *if* direction of the three statements in Proposition 35 does not hold in general, and this is where the extension-based perspective is different from the labelling-based one. For instance, the combination of **OO**, **UO**, **OU** and **UU** addition monotonicity does not imply $\not\mathbf{I}\not\mathbf{I}$ addition monotonicity. Indeed, it is possible define a semantics that satisfies **OO**, **UO**, **OU** and **UU** addition monotonicity but not $\not\mathbf{I}\not\mathbf{I}$ addition monotonicity.[1]

Using Proposition 34 we can derive results for $\not\mathbf{I}\mathbf{I}$, $\mathbf{I}\not\mathbf{I}$ and $\not\mathbf{I}\not\mathbf{I}$ addition and removal persistence under all the semantics we consider. Using Proposition 35 we can furthermore derive most of the results for $\not\mathbf{I}\mathbf{I}$, $\mathbf{I}\not\mathbf{I}$ and $\not\mathbf{I}\not\mathbf{I}$ addition and removal monotonicity. The following Propositions establish results for addition and removal monotonicity principles that do not follow using Proposition 35. A complete overview of these results can be found in Section 11.

[1] As an example, consider a semantics σ that, given that $F = (\{a, b\}, \{(a, a)\})$ and $F' = (\{a, b\}, \{(a, a), (a, b)\})$, satisfies $\mathcal{L}_\sigma(F) = \{\{(a, \mathbf{O}), (b, \mathbf{U})\}, \{(a, \mathbf{U}), (b, \mathbf{U})\}\}$ and $\mathcal{L}_\sigma(F') = \mathcal{L}_\sigma(F) \cup \{\{(a, \mathbf{I}), (b, \mathbf{O})\}\}$.

Proposition 36. *The grounded semantics satisfies* $\not{\mathbf{I}}\not{\mathbf{I}}$ *addition monotonicity and* $\not{\mathbf{I}}\mathbf{I}$ *removal monotonicity.*

Proof. This follows from the fact that the grounded semantics is a unique status semantics and satisfies $\not{\mathbf{I}}\not{\mathbf{I}}$ addition persistence and $\not{\mathbf{I}}\mathbf{I}$ removal persistence. □

Proposition 37. *The complete semantics satisfies* $\not{\mathbf{I}}\mathbf{I}$ *removal monotonicity.*

Proof. This follows from the fact that the complete semantics satisfies **OI** removal monotonicity (Proposition 30) together with the fact that, if x attacks y, there is no complete labelling in which x is labelled **U** and y is labelled **I**. □

8. Discussion

In this section, we summarise and discuss the results obtained so far. Table 5 provides an overview of all the labelling-based robustness principles. Table 6 provides a similar overview for the extension-based robustness principles. Note that Table 5 is a duplicate of the tables included in earlier sections, while the results in Table 6 follow from the results on labelling-based principles via Propositions 34 and 35, and otherwise from Propositions 36 and 37.

A number of general insights can be extracted from these tables. If we look at the extension-based robustness principles we see that the semi-stable, ideal, and eager do not satisfy any robustness principle, that the preferred semantics satisfies one, the complete semantics three, the grounded semantics four, and the stable semantics five. Moreover, of the few principles satisfied, most are persistence principles. In the labelling-based setting things look very different. Here, almost half of the principles are satisfied. This

Table 5

Addition/removal persistence/monotonicity for labelling-based semantics

Addition persistence	OO	OU	UO	UU	OI	UI	IO	IU	II
Grounded	■	■	■	■			■		
Complete	■	■	■	■	■		■		
Preferred	■	■	■		■		■		
Stable	■	■	■	■	■	■	■	■	
Semi-stable									
Ideal									
Eager									

Removal persistence	OO	OU	UO	UU	OI	UI	IO	IU	II
Grounded	■	■	■		■	■		■	■
Complete	■	■	■		■	■		■	■
Preferred	■	■	■		■	■		■	■
Stable	■	■	■	■	■	■		■	■
Semi-stable						■		■	■
Ideal						■		■	■
Eager						■		■	■

Addition monotonicity	OO	OU	UO	UU	OI	UI	IO	IU	II
Grounded	■	■	■	■			■		
Complete	■	■	■				■		
Preferred									
Stable		■	■	■		■		■	
Semi-stable									
Ideal									
Eager									

Removal monotonicity	OO	OU	UO	UU	OI	UI	IO	IU	II
Grounded	■	■	■		■	■		■	■
Complete	■	■	■		■	■		■	■
Preferred						■		■	■
Stable		■	■	■		■		■	■
Semi-stable						■		■	■
Ideal						■		■	■
Eager						■		■	■

Table 6

Addition/removal persistence/monotonicity for extension-based semantics

	Addition persistence			Removal persistence		
	$\not\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\mathbf{I}$	$\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\mathbf{I}$	$\mathbf{I}\not\mathbf{I}$
Grounded	■				■	
Complete	■				■	
Preferred					■	
Stable	■	■	■	■	■	
Semi-stable						
Ideal						
Eager						

	Addition monotonicity			Removal monotonicity		
	$\not\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\mathbf{I}$	$\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\not\mathbf{I}$	$\not\mathbf{I}\mathbf{I}$	$\mathbf{I}\not\mathbf{I}$
Grounded	■				■	
Complete					■	
Preferred						
Stable						
Semi-stable						
Ideal						
Eager						

supports the intuition that the additional discriminative power of labellings is very useful in describing the robustness of semantics.

Dung's traditional admissibility-based semantics (grounded, complete, preferred and stable) all behave differently with respect to the desirable robustness principles, so these principles are useful to distinguish these semantics in this respect. However, our principles do not distinguish the semi-stable, ideal and eager semantics, as none of these semantics satisfy any of the desirable principles (extension-based or labelling-based). The stable semantics appears to be the best behaved one with respect to the extension-based persistence principles. The same holds for the labelling-based persistence principles. Note however, that all the labelling-based principles involving **U**-labelled arguments are trivially satisfied under the stable semantics.

Throughout this paper, we made a distinction between persistence and monotonicity principles. We can also combine the two principles. Let us say that a semantics satisfies XY *addition/removal robustness* whenever it satisfies both XY addition/removal persistence and XY addition/removal monotonicity. Thus, XY addition/removal robustness, if satisfied by a semantics σ, means that if in all σ labellings, the arguments x and y are labelled X and Y, then the set of σ labellings remains the same when adding or removing an attack between x and y. Tables 7 and 8 provide overviews of robustness under both the labelling-based and extension-based settings.

Note that for the grounded, ideal and eager semantics, which are unique status semantics, robustness coincides with persistence and monotonicity. The preferred, semi-stable, ideal and eager semantics violate all robustness principles, except for the **UI**, **IU**, **II** removal robustness principles, which are trivially satisfied. Furthermore, the results for extension-based robustness principles look quite different from the labelling-based robustness principles. Here we have that the grounded semantics satisfies $\not\mathbf{I}\not\mathbf{I}$ addition robustness and $\not\mathbf{I}\mathbf{I}$ removal robustness. Other than that, the complete semantics satisfies $\not\mathbf{I}\mathbf{I}$ removal robustness, but no further robustness principles are satisfied under any of the semantics we consider.

Table 7

Addition/removal robustness for labelling-based semantics

	Addition robustness								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	■	■	■	■			■		
Complete	■	■	■				■		
Preferred									
Stable		■	■	■		■		■	
Semi-stable									
Ideal									
Eager									

	Removal robustness								
	OO	**OU**	**UO**	**UU**	**OI**	**UI**	**IO**	**IU**	**II**
Grounded	■	■	■		■	■		■	■
Complete	■	■	■		■	■		■	■
Preferred						■		■	■
Stable		■	■	■		■		■	■
Semi-stable						■		■	■
Ideal						■		■	■
Eager						■		■	■

Table 8

Addition/removal robustness for extension-based semantics

	Addition robustness		
	$\not{I}\not{I}$	$\not{I}I$	$I\not{I}$
Grounded	■		
Complete			
Preferred			
Stable			
Semi-stable			
Ideal			
Eager			

	Removal robustness		
	$\not{I}\not{I}$	$\not{I}I$	$I\not{I}$
Grounded		■	
Complete		■	
Preferred			
Stable			
Semi-stable			
Ideal			
Eager			

9. Future work

The results of the robustness principles introduced and analyzed in this paper give rise to three main lines of further research. First, the analysis can be extended to other semantics, such as non-admissible-based semantics. Second, the set of principles discussed in this chapter can be further extended. Third, relations between principles can be studied, such as relations between the robustness principles introduced in this paper and principles discussed in the literature like directionality and SCC decomposability. We consider these three lines of research below.

9.1. Semantics

The first topic for further research is to verify whether *other semantics* proposed in the literature satisfy our new principles. In some cases this will be easy. For example, given the construction of prudent semantics [33], it is easy to show that most of our principles are not satisfied. However, for naive-based semantics such as the Stage, Stage2 and CF2 semantics [5,43,61], it is often more difficult to see whether our robustness principles are satisfied or not. This is analogous to the case for many other principles, where for naive-based semantics often larger counter-examples are given [60]. Recently, a range of new semantics based on the notion of *weak admissibility* have been introduced [15]. Dauphin *et al.* [35] provide a principle-based analysis of these semantics, and also define the closely related *qualified* and *semi-qualified* semantics. Whether these new semantics satisfy our robustness principles is also an open question.

The second topic for further research is the consideration of other semantic frameworks such as *other labelling conventions*. As we have shown, there is a clear distinction between the extension-based results in Table 5 and the labelling-based results in Table 6. Whereas for extension-based semantics very few principles are satisfied, many more principles are satisfied for labelling-based semantics. This can be readily explained: the distinction between out and undecided arguments makes the principles more precise. However, it also raises a question: if we adopt another way to describe the semantics, for example using four valued labelling functions, would that further increase the precision of the analysis? We can invert the question: which set of truth values can be defined such that we have maximal persistence and monotonicity principles?

The third topic of further research is to use the robustness principles to guide the search for new semantics, as in general suggested by van der Torre and Vesic [60]. However, whereas robustness may be desirable in some applications, it may be seen as a drawback in others. For example, the most robust semantics is one in which the set of extensions is always the same. In other words, the search for new semantics may require to make additional requirements explicit, for example in terms of additional principles. Moreover, we need to decide which of our robustness principles can play the role of desiderata in such a search, maybe in the form of meta-principles.

Finally, a fourth topic for further research is to consider robustness for ranking semantics, or other semantics functions. Likewise, one can extend the definition of robustness and the analysis to more expressive argumentation frameworks, for example expressing collective or set attack, higher-order attack, support relation in bipolar argumentation frameworks, or abstract dialectical frameworks.

9.2. Principles

The fifth topic for further research is to define principles by *weakening or adapting the existing principles*. This can be done to find principles which are more discriminative, or that correspond to particular intuitions. For example, since we are typically interested only in the accepted arguments, we may require that a labelling does not have to be exactly the same, only the in-part needs to remain unchanged. As another example, since self-attacks are known to cause problems in many examples, we may consider only frameworks that do not contain self-attacks, or that more generally do not contain odd loops.

9.3. Relations between principles

The sixth topic for further research is to study logical relations among the principles, independently of the semantics. For example, we may verify whether one property implies another one. In particular, the relation between our principles and existing ones needs to be further studied. The most obvious ones are directionality, and SCC decomposability.

The seventh topic for further research is to consider sets of principles. For example, as common in social choice, there may be sets of principles such that there are no semantics satisfying all of them, or there may be sets of principles such that there is exactly one semantics satisfying all of them. We are not aware of such impossibility or characterisation results in the literature on formal argumentation.

10. Related work

The robustness principles investigated in this paper build on earlier work by Boella *et al.* [22,23], who studied *refinement* and *abstraction* principles, which are conditions under which the evaluation of an argumentation framework remains unchanged when an attack is added or removed. However, they consider only semantics that yield a *single* extension or labelling. The results we obtained for addition and removal persistence under the grounded semantics coincide with theirs.

Our principles are related to the so called *explicit conflict conjecture* which was introduced by Baumann *et al.* [17] in the investigation of the notion of *compact argumentation frameworks*. These are argumentation frameworks that are minimal in the sense that no argument can be removed from the argumentation framework without affecting the set of extensions. For the definition, let $F = (A, \rightsquigarrow)$ be an argumentation framework and let σ be a semantics. Two arguments $x, y \in A$ are said to be in *conflict* with respect to F and σ if no σ extension of F contains both x and y. If x and y are in conflict with respect to F and σ then this conflict is called an *explicit conflict* if $x \rightsquigarrow y$ and an *implicit conflict* if $x \not\rightsquigarrow y$. The explicit conflict conjecture is said to hold for a semantics σ if, for every argumentation framework $F = (A, \rightsquigarrow)$ there exists an argumentation framework $F' = (A, \rightsquigarrow')$ such that the σ extensions of F and F' coincide, while all the conflicts with respect to F' and σ are explicit. While this principle was initially conjectured to hold for the stable semantics [17], in a later publication it is actually shown not to hold for the stable semantics, nor for the preferred or semi-stable semantics [16]. The relationship with our principles is the following. If a semantics σ satisfies ⋈ robustness, then between any pair of arguments in an implicit conflict, we can add an attack without causing a change in the set of extensions or labellings. Hence, for every argumentation framework (possibly containing implicit conflicts) we can construct an argumentation framework with the same set of σ extensions that contains no implicit conflicts. Hence, ⋈ robustness implies the explicit conflict conjecture. Table 8 shows that, apart from the grounded semantics (for which the explicit conflict conjecture can easily be shown to hold) none of the semantics we consider satisfy ⋈ robustness. This is consistent with the findings of Baumann *et al.* [16].

Cayrol *et al.* [32] studied the impact on the evaluation of an argumentation framework when new arguments and attacks are added. The main difference with our approach is that they focus on different ways in which the evaluation changes, rather than conditions under which the evaluation does not change. Some types of change that they define are changes leading to a larger, unique, or smaller set of extensions, changes that are monotonic (every extension of the old argumentation framework is included in an extension of the changed argumentation framework) and changes that are monotonous with respect to an argument (every argument included in an extension of the old argumentation framework is also included in an extension of the changed argumentation framework). They then define conditions under which the addition of an argument to an argumentation framework leads to these types of change.

The notion of *strong equivalence* in abstract argumentation deals with an alternative perspective on robustness in abstract argumentation. Given two argumentation frameworks $F = (A, \rightsquigarrow)$ and $F' = (A', \rightsquigarrow')$, let $F \cup F'$ denote the argumentation framework $(A \cup A', \rightsquigarrow \cup \rightsquigarrow')$. Two argumentation frameworks F and G are strongly equivalent with respect to a semantics σ if, for every argumentation framework H, the σ extensions of the argumentation frameworks $F \cup H$ and $G \cup H$ are equivalent. A natural question to ask is: in what kind of way can an argumentation framework F be changed into another argumentation framework F' such that F and F' are strongly equivalent? Oikarinen and Woltran [54] address this question by characterising strong equivalence under various semantics using the notion of *kernel*. A kernel of an argumentation framework F is another argumentation framework that is the same as F except that certain attacks are deleted according to a given syntactic or graph structural criterion. A kernel characterises strong equivalence under a semantics σ if it holds, for any two argumentation frameworks F and G, that F and G are strongly equivalent whenever the kernels of F and G are equivalent. An example is the kernel where every attack is deleted between pairs of distinct arguments where both arguments are self-attacking, which characterises strong equivalence under the complete semantics. A different way of stating this result is that, under the complete semantics, the addition and removal of attacks between two self-attacking arguments of an argumentation framework F has no effect on the set of complete extensions of F, even if F is extended with additional arguments and attacks. This result (and similar results for other semantics) may be contrasted with the results we obtained in this paper, which also imply that under certain conditions, the addition or removal of an attack between two arguments x and y has no effect on the set of extensions or labellings. The difference is that we use conditions that refer to the status of x and y, while strong equivalence results refer to syntactic or graph structural conditions and not on the status of arguments. Further research in this direction includes the study of various other notions of equivalence, which deal with deletions, updates or restricted forms of change [7,8]. Baumann furthermore studied equivalence notions for labelling-based semantics [9], and Baumann and Spanring studied equivalence notions in the unrestricted case (i.e., not assuming finite argumentation frameworks) [18]. A recent overview of results can be found in [10].

Bistarelli *et al.* [21] study so called *local-expansion invariant operators* for abstract argumentation frameworks. These operators are functions that take an argumentation framework as input and return the same argumentation framework except that zero or more attacks are added. They then determine conditions under which such operators are invariant with respect to a semantics σ. An operator is invariant with respect to a semantics σ precisely when the set of σ extensions remains unchanged after applying the operator. They focus only on conflict-freeness and admissibility but plan to study further semantics in future work.

Sakama studied counterfactuals in the context of abstract argumentation [58]. Such counterfactuals have the form $\alpha \Box\!\!\rightarrow \beta$ (meaning "if α were true then β would be true") where α and β are literals of the form $\mathbf{I}(x)$ or $\mathbf{O}(x)$ for some argument x. A counterfactual $\alpha \Box\!\!\rightarrow \beta$ is entailed by an argumentation framework F and semantics σ if the change of F represented by α leads to the truth of β in all σ labellings of F. Here, the premise $\mathbf{O}(x)$ represents the addition of a new argument attacking x, while $\mathbf{I}(x)$ represents the removal of all attacks pointing to x. In this setting, the persistence and monotonicity principles translate to properties of the set of counterfactuals entailed by an argumentation framework. Similarly, Rienstra defined a form of meta-inference for abstract argumentation frameworks [55]. Formally, an argumentation framework F and semantics σ defines a relation $|\!\!\models^F_\sigma$ between so called *interventions* and *consequences*. An intervention is a set of literals of the form $\mathbf{O}(x)$ or $\neg\mathbf{I}(x)$ that represent, respectively, the addition of a new argument attacking x or of a new self-attacking argument attacking x. A formula ϕ is a consequence of an intervention Φ (written $\Phi \;|\!\!\models^F_\sigma \phi$) if the change represented by Φ leads to the truth of ϕ in all σ labellings of F. Like in Sakama's approach, the principles defined in this paper for a semantics σ translate to properties that characterise the behaviour of the inference relation $|\!\!\models^F_\sigma$. Rienstra considers a number of common inference rules such as *Cautious Monotonicity* and *Cut*. Cautious Monotonicity states that, if $\Phi \;|\!\!\models^F_\sigma \alpha$ and $\Phi \;|\!\!\models^F_\sigma \phi$, then $\Phi \cup \{\alpha\} \;|\!\!\models^F_\sigma \phi$. Cut states that, if $\Phi \;|\!\!\models^F_\sigma \alpha$ and $\Phi \cup \{\alpha\} \;|\!\!\models^F_\sigma \phi$, then $\Phi \;|\!\!\models^F_\sigma \phi$. While these are properties that are generally considered desirable, Rienstra showed that they are not satisfied under all semantics. For example, Cautious Monotonicity does not hold under the preferred semantics. This follows from the violation of **IO** addition monotonicity under the preferred semantics (Example 33). Similarly, failure of Cut under the semi-stable semantics follows from the failure of **IO** addition persistence (Example 11).

Much work has been done in the past decade on what is often referred to as *dynamics* of argumentation. What unifies this work is its motivation, namely that argumentation is a dynamic process, which means that argumentation frameworks are not static entities but evolve over time. The current paper extends this line of work, as it contributes to the understanding of how admissibility-based semantics behave when the argumentation framework changes. There is a large amount of further work on dynamics of argumentation. For example, the question of whether and how an argumentation framework can be changed in order to enforce some result, such as the acceptance of a given set of arguments, is referred to as the *enforcement problem* [11,12,20,39,52,53,62]. Change in argumentation has furthermore been related to (or modelled using tools of) theories of belief change, often taking an AGM-inspired approach [13,14,25,34,36–38,45,48–50,57,59]. Computational issues in a dynamic setting have also been studied, such as efficiently (re)computing extensions after an argumentation framework changes [1,6,46]. In general, we believe that the robustness principles studied in this paper are relevant to much of the research on dynamics of argumentation. In the introduction we already discussed ways in which earlier work on robustness principles has been applied in the design of algorithms for the enforcement problem [53] and incomplete argumentation frameworks [51].

11. Summary and conclusion

We have defined a set of general principles for argumentation semantics expressing robustness of labellings and extensions against the addition and removal of attacks, and investigated all the admissibility-based semantics with respect to these principles. The results we obtain provide insight into the behaviour of these semantics in a dynamic context. For example, while the grounded, complete, preferred and stable semantics satisfy some of the desirable robustness principles, the semi-stable, eager and ideal semantics satisfy none. Furthermore, Dung's traditional semantics (i.e., grounded, complete, preferred and stable) all satisfy different sets of robustness principles, so these principles are useful to distinguish these semantics in this respect.

The robustness principles we investigate are also useful in the design of algorithms. As we have discussed, addition and removal persistence properties have already been applied in the design of algorithms. For instance, knowledge about whether the addition or removal of an attack has an effect on the evaluation of an argumentation framework may be used to optimise search problems connected with the enforcement problem in argumentation [53]. In this context an interesting observation is that the semi-stable, ideal and eager semantics do not satisfy any useful robustness principle, which implies that these semantics cannot benefit from such optimisations, because there are no conditions under robustness against addition or removal of attacks is guaranteed.

We discussed a number of directions for future research, such as extending our analysis by weakening or adapting the robustness principles, and studying relationships between our robustness principles and other principles studied in the literature. We may furthermore consider robustness in the context of more expressive argumentation formalisms, such as frameworks supporting collective or set attacks, higher-order attacks, support relations (bipolar argumentation frameworks) or abstract dialectical frameworks. Finally, we focused in this paper only on admissibility-based semantics. We plan to extend our investigation of robustness to non-admissibility-based semantics in a follow-up publication.

Acknowledgement

Tjitze Rienstra is supported by the Deutsche Forschungsgemeinschaft (grant KE 1686/3-1). Leendert van der Torre is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 690974 for the project MIREL (MIning and REasoning with Legal texts). Beishui Liao is supported by the Convergence Research Project for Brain Research and Artificial Intelligence, Zhejiang University, and the Natural Science Foundation of Zhejiang Province of China under Grant No. LY20F030014.

References

[1] G. Alfano, S. Greco and F. Parisi, An efficient algorithm for skeptical preferred acceptance in dynamic argumentation frameworks, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, Macao, China, August 10–16, 2019, S. Kraus, ed., ijcai.org, 2019, pp. 18–24. doi:10.24963/ijcai.2019/3.

[2] L. Amgoud and J. Ben-Naim, Ranking-based semantics for argumentation frameworks, in: *Scalable Uncertainty Management – 7th International Conference, SUM 2013. Proceedings*, Washington, DC, USA, September 16–18, 2013, W. Liu, V.S. Subrahmanian and J. Wijsen, eds, Lecture Notes in Computer Science, Vol. 8078, Springer2013, pp. 134–147. ISBN 978-3-642-40380-4. doi:10.1007/978-3-642-40381-1_11.

[3] O. Arieli, Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks, *J. Appl. Log.* **13**(4) (2015), 582–604. doi:10.1016/j.jal.2015.03.005.

[4] P. Baroni and M. Giacomin, On principle-based evaluation of extension-based argumentation semantics, *Artificial Intelligence* **171**(10–15) (2007), 675–700. doi:10.1016/j.artint.2007.04.004.

[5] P. Baroni, M. Giacomin and G. Guida, SCC-recursiveness: A general schema for argumentation semantics, *Artificial Intelligence* **168**(1–2) (2005), 162–210. doi:10.1016/j.artint.2005.05.006.

[6] P. Baroni, M. Giacomin and B. Liao, On topology-related properties of abstract argumentation semantics. a correction and extension to dynamics of argumentation systems: A division-based method, *Artificial Intelligence* **212** (2014), 104–115. doi:10.1016/j.artint.2014.03.003.

[7] R. Baumann, Normal and strong expansion equivalence for argumentation frameworks, *Artificial Intelligence* **193** (2012), 18–44. doi:10.1016/j.artint.2012.08.004.

[8] R. Baumann, Context-free and context-sensitive kernels: Update and deletion equivalence in abstract argumentation, in: *ECAI 2014–21st European Conference on Artificial Intelligence, 18–22 August 2014, Prague, Czech Republic – Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, T. Schaub, G. Friedrich and B. O'Sullivan, eds, Frontiers in Artificial Intelligence and Applications, Vol. 263, IOS Press, 2014, pp. 63–68. ISBN 978-1-61499-418-3. doi:10.3233/978-1-61499-419-0-63.

[9] R. Baumann, Characterizing equivalence notions for labelling-based semantics, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016*, Cape Town, South Africa, April 25–29, 2016, C. Baral, J.P. Delgrande and F. Wolter, eds, AAAI Press, 2016, pp. 22–32. http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12836. ISBN 978-1-57735-755-1.

[10] R. Baumann, On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability, *IfCoLog Journal of Logics and their Applications* **4**(8) (2017), 2779–2886. http://www.collegepublications.co.uk/admin/download.php?ID=ifcolog00017.

[11] R. Baumann, What does it take to enforce an argument? Minimal change in abstract argumentation, in: *ECAI 2012–20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track*, Montpellier, France, August 27–31 , 2012, Frontiers in Artificial Intelligence and Applications, Vol. 242, IOS Press, pp. 127–132.

[12] R. Baumann and G. Brewka, Expanding argumentation frameworks: Enforcing and monotonicity results, in: *COMMA 2010*, Desenzano del Garda, Italy, September 8–10, 2010, Frontiers in Artificial Intelligence and Applications, Vol. 216, IOS Press, 2010, pp. 75–86.

[13] R. Baumann and G. Brewka, AGM meets abstract argumentation: Expansion and revision for dung frameworks, in: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires*, Argentina, July 25–31, 2015, Q. Yang and M.J. Wooldridge, eds, AAAI Press, 2015, pp. 2734–2740. http://ijcai.org/Abstract/15/387. ISBN 978-1-57735-738-4.

[14] R. Baumann and G. Brewka, Extension removal in abstract argumentation – an axiomatic approach, in: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, Honolulu, Hawaii, USA, January 27–February 1, 2019, AAAI Press, 2019, pp. 2670–2677. ISBN 978-1-57735-809-1. doi:10.1609/aaai.v33i01.33012670.

[15] R. Baumann, G. Brewka and M. Ulbricht, Revisiting the foundations of abstract argumentation – semantics based on weak admissibility and weak defense, in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, New York, NY, USA, February 7–12, 2020, AAAI Press, pp. 2742–2749. https://aaai.org/ojs/index.php/AAAI/article/view/5661. ISBN 978-1-57735-823-7.

[16] R. Baumann, W. Dvořák, T. Linsbichler, C. Spanring, H. Strass and S. Woltran, On rejected arguments and implicit conflicts: The hidden power of argumentation semantics, *Artif. Intell.* **241** (2016), 244–284. doi:10.1016/j.artint.2016.09.004.

[17] R. Baumann, W. Dvorák, T. Linsbichler, H. Strass and S. Woltran, Compact argumentation frameworks, in: *ECAI 2014–21st European Conference on Artificial Intelligence, 18–22 August 2014, Prague, Czech Republic – Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, T. Schaub, G. Friedrich and B. O'Sullivan, eds, Frontiers in Artificial Intelligence and Applications, Vol. 263, IOS Press, 2014, pp. 69–74. ISBN 978-1-61499-418-3. doi:10.3233/978-1-61499-419-0-69.

[18] R. Baumann and C. Spanring, A study of unrestricted abstract argumentation frameworks, in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, Melbourne, Australia, August 19–25, 2017, C. Sierra, ed., ijcai.org, 2017, pp. 807–813. ISBN 978-0-9992411-0-3. doi:10.24963/ijcai.2017/112.

[19] P. Besnard, V. David, S. Doutre and D. Longin, Subsumption and incompatibility between principles in ranking-based argumentation, in: *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017*, Boston, MA, USA, November 6–8, 2017, IEEE Computer Society, pp. 853–859. ISBN 978-1-5386-3876-7. doi:10.1109/ICTAI.2017.00133.

[20] P. Bisquert, C. Cayrol, F.D. de Saint-Cyr and M.-C. Lagasquie-Schiex, Enforcement in argumentation is a kind of update, in: *Lecture Notes in Computer Science*, W. Liu, V.S. Subrahmanian and J. Wijsen, eds, Vol. 8078, Springer, 2013, pp. 30–43. ISBN 978-3-642-40380-4. doi:10.1007/978-3-642-40381-1.

[21] S. Bistarelli, F. Santini and C. Taticchi, Local expansion invariant operators in argumentation semantics, in: *Dynamics, Uncertainty and Reasoning, The Second Chinese Conference on Logic and Argumentation, CLAR 2018*, Hangzhou, China, 16–17 June 2018, B. Liao, T. Ågotnes and Y.N. Wáng, eds, Springer, 2018, pp. 45–62. ISBN 978-981-13-7790-7. doi:10.1007/978-981-13-7791-4_3.

[22] G. Boella, S. Kaci and L. van der Torre, Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension, in: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 10th European Conference, ECSQARU 2009, Proceedings*, Verona, Italy, July 1–3, 2009, LNCS, Vol. 5590, Springer, 2009, pp. 107–118.

[23] G. Boella, S. Kaci and L. van der Torre, Dynamics in argumentation with single extensions: Attack refinement and the grounded extension (extended version), in: *Argumentation in Multi-Agent Systems, 6th International Workshop, ArgMAS 2009, Revised Selected and Invited Papers*, Budapest, Hungary, May 12, 2009, LNCS, Vol. 6057, Springer, 2009, pp. 150–159.

[24] E. Bonzon, J. Delobelle, S. Konieczny and N. Maudet, A comparative study of ranking-based semantics for abstract argumentation, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, February 12–17, 2016, D. Schuurmans and M.P. Wellman, eds, AAAI Press, 2016, pp. 914–920. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12465. ISBN 978-1-57735-760-5.

[25] R. Booth, S. Kaci, T. Rienstra and L. van der Torre, *A Logical Theory About Dynamics in Abstract Argumentation*, W. Liu, V.S. Subrahmanian and J. Wijsen, eds, Lecture Notes in Computer Science, Vol. 8078, Springer, 2013, pp. 148–161. ISBN 978-3-642-40380-4. doi:10.1007/978-3-642-40381-1.

[26] M. Caminada, Comparing two unique extension semantics for formal argumentation: Ideal and eager, in: *The 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007)*, 2007, pp. 81–87.

[27] M. Caminada, Rationality postulates: Applying argumentation theory for non-monotonic reasoning, *IfCoLog Journal of Logics and their Applications* **4**(8) (2017), 2707–2732. http://www.collegepublications.co.uk/admin/download.php?ID=ifcolog00017.

[28] M. Caminada, On the issue of reinstatement in argumentation, in: *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Proceedings*, Liverpool, UK, September 13–15, 2006, LNCS, Vol. 4160, Springer, pp. 111–123.

[29] M. Caminada and L. Amgoud, On the evaluation of argumentation formalisms, *Artif. Intell.* **171**(5–6) (2007), 286–310. doi:10.1016/j.artint.2007.02.003.

[30] M.W. Caminada, W.A. Carnielli and P.E. Dunne, Semi-stable semantics, *Journal of Logic and Computation* **22**(5) (2011), 1207–1254. doi:10.1093/logcom/exr033.

[31] M.W.A. Caminada and D.M. Gabbay, A logical account of formal argumentation, *Studia Logica* **93**(2–3) (2009), 109–145. doi:10.1007/s11225-009-9218-x.

[32] C. Cayrol, F.D. de Saint-Cyr and M.-C. Lagasquie-Schiex, Change in abstract argumentation frameworks: Adding an argument, *Journal of Artificial Intelligence Research* **38** (2010), 49–84. doi:10.1613/jair.2965.

[33] S. Coste-Marquis, C. Devred and P. Marquis, Prudent semantics for argumentation frameworks, in: *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, Hong Kong, China, 14–16 November 2005, IEEE Computer Society, 2005, pp. 568–572.

[34] S. Coste-Marquis, S. Konieczny, J.-G. Mailly and P. Marquis, On the revision of argumentation systems: Minimal change of arguments statuses, in: *Principles of Knowledge Representation and Reasoning: The 14th International Conference, KR 2014*, Vienna, Austria, July 20–24, 2014, C. Baral, G. De Giacomo and T. Eiter, eds, AAAI Press, 2014.

[35] J. Dauphin, T. Rienstra and L. van der Torre, A principle-based analysis of weakly admissible semantics, in: *Computational Models of Argument – Proceedings of COMMA 2020*, Perugia, Italy, September 4–11, 2020, H. Prakken, S. Bistarelli, F. Santini and C. Taticchi, eds, Frontiers in Artificial Intelligence and Applications, Vol. 326, IOS Press, 2020, pp. 167–178. ISBN 978-1-64368-106-1. doi:10.3233/FAIA200502.

[36] F.D. de Saint-Cyr, P. Bisquert, C. Cayrol and M. Lagasquie-Schiex, Argumentation update in YALLA (Yet Another Logic Language for Argumentation), *Int. J. Approx. Reason.* **75** (2016), 57–92. doi:10.1016/j.ijar.2016.04.003.
[37] M. Diller, A. Haret, T. Linsbichler, S. Rümmele and S. Woltran, An extension-based approach to belief revision in abstract argumentation, *Int. J. Approx. Reason.* **93** (2018), 395–423. doi:10.1016/j.ijar.2017.11.013.
[38] S. Doutre, A. Herzig and L. Perrussel, A dynamic logic framework for abstract argumentation, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*, Vienna, Austria, July 20–24, 2014, C. Baral, G. De Giacomo and T. Eiter, eds, AAAI Press, 2014. http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7993.
[39] S. Doutre and J. Mailly, Constraints and changes: A survey of abstract argumentation dynamics, *Argument & Computation* **9**(3) (2018), 223–248. doi:10.3233/AAC-180425.
[40] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* **77**(2) (1995), 321–358. doi:10.1016/0004-3702(94)00041-X.
[41] P.M. Dung, P. Mancarella and F. Toni, Computing ideal sceptical argumentation, *Artif. Intell.* **171**(10–15) (2007), 642–674. doi:10.1016/j.artint.2007.05.003.
[42] P.E. Dunne, W. Dvorák, T. Linsbichler and S. Woltran, Characteristics of multiple viewpoints in abstract argumentation, in: *Principles of Knowledge Representation and Reasoning: The 14th International Conference, KR 2014*, Vienna, Austria, July 20–24, 2014, C. Baral, G. De Giacomo and T. Eiter, eds, AAAI Press, 2014.
[43] W. Dvorák and S.A. Gaggl, Stage semantics and the SCC-recursive schema for argumentation semantics, *J. Log. Comput.* **26**(4) (2016), 1149–1202. doi:10.1093/logcom/exu006.
[44] W. Dvorák and C. Spanring, Comparing the expressiveness of argumentation semantics, *J. Log. Comput.* **27**(5) (2017), 1489–1521. doi:10.1093/logcom/exw008.
[45] M.A. Falappa, G. Kern-Isberner and G.R. Simari, Belief revision and argumentation theory, in: *Argumentation in Artificial Intelligence*, G.R. Simari and I. Rahwan, eds, Springer, 2009, pp. 341–360. ISBN 978-0-387-98196-3. doi:10.1007/978-0-387-98197-0_17.
[46] B.S. Liao, L. Jin and R.C. Koons, Dynamics of argumentation systems: A division-based method, *Artificial Intelligence* **175**(11) (2011), 1790–1814. doi:10.1016/j.artint.2011.03.006.
[47] T. Linsbichler, J. Pührer and H. Strass, A uniform account of realizability in abstract argumentation, in: *ECAI 2016–22nd European Conference on Artificial Intelligence, 29 August–2 September 2016, The Hague, The Netherlands – Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, G.A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum and F. van Harmelen, eds, Frontiers in Artificial Intelligence and Applications, Vol. 285, IOS Press, 2016, pp. 252–260. ISBN 978-1-61499-671-2. doi:10.3233/978-1-61499-672-9-252.
[48] M.O. Moguillansky, N.D. Rotstein, M.A. Falappa, A.J. García and G.R. Simari, Dynamics of knowledge in DeLP through argument theory change, *Theory Pract. Log. Program.* **13**(6) (2013), 893–957. doi:10.1017/S1471068411000603.
[49] M.O. Moguillansky and G.R. Simari, A generalized abstract argumentation framework for inconsistency-tolerant ontology reasoning, *Expert Syst. Appl.* **64** (2016), 141–168. doi:10.1016/j.eswa.2016.07.027.
[50] M.O. Moguillansky, R. Wassermann and M.A. Falappa, Inconsistent-tolerant base revision through argument theory change, *Log. J. IGPL* **20**(1) (2012), 154–186. doi:10.1093/jigpal/jzr029.
[51] A. Niskanen, D. Neugebauer, M. Järvisalo and J. Rothe, Deciding acceptance in incomplete argumentation frameworks, in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, AAAI Press, 2020.
[52] A. Niskanen, J.P. Wallner and M. Järvisalo, Optimal status enforcement in abstract argumentation, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, New York, NY, USA, 9–15 July 2016, S. Kambhampati, ed., IJCAI/AAAI Press, 2016, pp. 1216–1222. http://www.ijcai.org/Abstract/16/176. ISBN 978-1-57735-770-4.
[53] A. Niskanen, J.P. Wallner and M. Järvisalo, Extension enforcement under grounded semantics in abstract argumentation, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, Tempe, Arizona, 30 October–2 November 2018, M. Thielscher, F. Toni and F. Wolter, eds, AAAI Press, 2018, pp. 178–183. https://aaai.org/ocs/index.php/KR/KR18/paper/view/18040. ISBN 978-1-57735-803-9.
[54] E. Oikarinen and S. Woltran, Characterizing strong equivalence for argumentation frameworks, *Artificial Intelligence* **175**(14–15) (2011), 1985–2009. doi:10.1016/j.artint.2011.06.003.
[55] T. Rienstra, Argumentation in Flux – Modelling Change in the Theory of Argumentation, PhD thesis, University of Luxembourg/University of Montpellier II, 2014.
[56] T. Rienstra, C. Sakama and L. van der Torre, Persistence and monotony properties of argumentation semantics, in: *Theory and Applications of Formal Argumentation – Third International Workshop, TAFA 2015, Revised Selected Papers*, Buenos Aires, Argentina, July 25–26, 2015, E. Black, S. Modgil and N. Oren, eds, Lecture Notes in Computer Science, Vol. 9524, Springer, 2015, pp. 211–225. ISBN 978-3-319-28459-0. doi:10.1007/978-3-319-28460-6_13.
[57] N.D. Rotstein, M.O. Moguillansky, M.A. Falappa, A.J. García and G.R. Simari, Argument theory change: Revision upon warrant, in: *COMMA 2008*, Toulouse, France, May 28–30, 2008, Frontiers in Artificial Intelligence and Applications, Vol. 172, IOS Press, 2008, pp. 336–347.

[58] C. Sakama, Counterfactual reasoning in argumentation frameworks, in: *Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish*, Highlands, UK, September 9–12, 2014, Frontiers in Artificial Intelligence and Applications, Vol. 266, IOS Press, 2014, pp. 385–396.

[59] M. Snaith and C. Reed, Argument revision, *J. Log. Comput.* **27**(7) (2017), 2089–2134. doi:10.1093/logcom/exw028.

[60] L. van der Torre and S. Vesic, The principle-based approach to abstract argumentation semantics, *IfCoLog Journal of Logics and their Applications* **4**(8) (2017), 2735–2779. http://www.collegepublications.co.uk/admin/download.php?ID=ifcolog00017.

[61] B. Verheij, Two approaches to dialectical argumentation: Admissible sets and argumentation stages, in: *The Eighth Dutch Conference on Artificial Intelligence (NAICO96)*, 1996, pp. 357–368.

[62] J.P. Wallner, A. Niskanen and M. Järvisalo, Complexity results and algorithms for extension enforcement in abstract argumentation, *J. Artif. Intell. Res.* **60** (2017), 1–40. doi:10.1613/jair.5415.