2020-01-28

Beyond reasonable doubt: a proposal for undecidedness blocking in abstract argumentation

Pierpaolo Dondio
Technological University Dublin

Luca Longo
Technological University Dublin, luca.longo@tudublin.ie

Recommended Citation

Dondio, Pierpaolo and Longo, Luca. 'Beyond Reasonable Doubt: A Proposal for Undecidedness Blocking in Abstract Argumentation'. 1 Jan. 2019 : 123 – 135. DOI: 10.3233/IA-190030

Intelligenza Artificiale 13 (2019) 123–135
DOI 10.3233/IA-190030
IOS Press

Beyond reasonable doubt: A proposal for undecidedness blocking in abstract argumentation

Pierpaolo Dondio* and Luca Longo
School of Computer Science, Technological University Dublin, Dublin, Ireland

Abstract. In Dung's abstract semantics, the label undecided is always propagated from the attacker to the attacked argument, unless the latter is also attacked by an accepted argument. In this work we propose *undecidedness blocking* abstract argumentation semantics where the undecided label is confined to the strong connected component where it was generated and it is not propagated to the other parts of the argumentation graph. We show how u*ndecidedness blocking* is a fundamental reasoning pattern absent in abstract argumentation but present in similar fashion in the *ambiguity blocking* semantics of Defeasible logic, in the *beyond reasonable doubt* legal principle or when someone gives someone else the *benefit of the doubt*. The resulting semantics, called SCC-void semantics, are defined using an SCC-recursive schema. The semantics are conflict-free and non-admissible, but they incorporate a more relaxed defence-based notion of admissibility. They allow reinstatement and they credulously accept what the corresponding Dung's complete semantics accepts at least credulously.

Keywords: Abstract argumentation semantics, ambiguity blocking, standard of proofs, undecidedness

1. Introduction

Abstract argumentation is a framework for non-monotonic reasoning where conclusions are reached by evaluating arguments and their conflict relations. Abstract argumentation is centred on the notion of argumentation framework [11, 22], a directed graph where nodes represent arguments and links represent attack relations defined over arguments.

One of the main tasks of abstract argumentation is the computation of the acceptability status of arguments. This is achieved by the application of an argumentation semantics, a set of postulates used to identify the sets of arguments, called extensions, which successfully survive the conflicts encoded in the attack relations. In the labelling approach proposed in [4], the effect of an argumentation semantics is to assign to each argument a label *in, out* or *undec*. This means that an argument can respectively be accepted, rejected or deemed undecided. The *undec* label represents a situation in which the semantics has no reasons to accept or reject an argument.

In this paper the definition of a new family of abstract semantics called SCC-void semantics is explored. For each complete semantics, it is possible to define a SCC-void version of it. SCC-void semantics are conflict-free and non-admissible, but they still employ a defence-based relaxed notion of admissibility. They allow reinstatement and generate extensions that are supersets of the extensions generated by the corresponding complete semantics. Since the paper introduces *yet another* semantics, it is important to consider why many semantics have been proposed and where our new semantics fit in terms of motivation and originality.

*Corresponding author: Pierpaolo Dondio, School of Computer Science, Technological University Dublin, Kevin Street, Dublin 2, Ireland. E-mail: pierpaolo.dondio@TUDublin.ie.

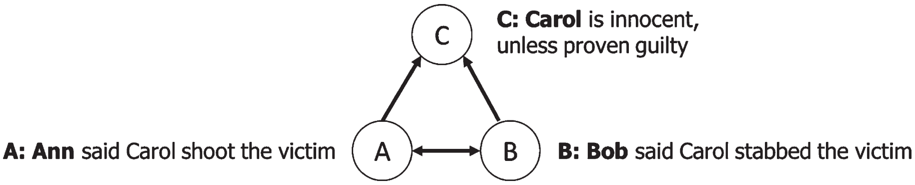
C: Carol is innocent, unless proven guilty
C
A: Ann said Carol shoot the victim
A
B
B: Bob said Carol stabbed the victim

Fig. 1. The floating assignment in a legal context.

The variety of semantics that exists in the literature of argumentation is due to at least two reasons: (i) different semantics might model different principles and (ii) the same principle can be modelled in different ways and consequently it can lead to different strategies for acceptability. For instance, Dung's complete semantics [11] are all based on the notion of reinstatement; however, they differ in the way this notion is used across them. For example, on one hand, under the grounded semantics an argument can be reinstated only by a different accepted argument. On the other hand, under the preferred semantics an argument can indeed reinstate itself by attacking its attackers.

Baroni noticed how different semantics reflect different intuitions and are meant to satisfy specific properties and/or fulfil some desired behaviours in problematic examples [24]. Given the same argumentation framework, it is not unusual that two abstract semantics differ even strongly on the acceptability status of arguments. For instance, if we consider the argumentation graph depicted in Fig. 1, known as the *floating assignment* example, argument C is undecided under the grounded semantics but rejected (sceptically) under preferred or stable semantics.

This behaviour is not a surprise. Each semantics offers a formal model of some reasoning patterns that could fit some contexts and be questionable in others. However, in abstract argumentation arguments are atomic units and no information about their internal structure or the context are provided. As a consequence, several interpretations of the same graph are possible. While, at the abstract level, there is no reason to prefer one semantics over another, it is only when contextual information about the situation is considered that one could consider one semantics a better fit than another. For instance, few authors [19, 23] underline how grounded semantics might be more appropriate for epistemic reasoning while preferred semantics for practical reasoning. However, an argumentation graph does not encode any information about the epistemic or practical nature of the discussion modelled.

Baroni [24] noticed how, included in the contextual information required to meaningfully applied a semantics, there are also design choices such as the way arguments are constructed and attack relations identified as well as choices that directly interact with argumentation semantics.

The new semantics introduced in this paper follow the same line of thought: they need contextual information in order to evaluate their appropriateness to the context.

The motivation for these new semantics is to model a fundamental reasoning mechanism absent in current abstract semantics. This mechanism, that we call *undecidedness blocking*, is inspired by the notion of ambiguity blocking semantics [14, 29], by the *beyond reasonable doubt* legal principle or by common decision-making patterns such as the *benefit of the doubt*. The genesis of such new semantics starts from a reconsideration of the way the undecided label is propagated onto an argumentation framework. Under any of Dung's complete semantics, an argument a attacked by an undecided argument is never accepted. It is either undecided or, if another accepted arguments is also attacking a, it is rejected. Therefore the undecided label is always propagated from the attacker to the attacked argument in the absence of accepted attackers. Using a terminology belonging to defeasible logic, Governatori [14] observed how Dung's complete semantics are all ambiguity propagating.

In this work we prefer to speak about *undecidedness* propagation (or blocking) rather than ambiguity propagation (or blocking). Undecidedness is not the same as ambiguity and we also intend to detach ourselves from the notion of ambiguity of Defeasible Logic, as we explain later in this section. An agent blocking undecidedness would prevent the undecided arguments to affect other parts of a reasoning process. An attack from such arguments, for instance, would not be effective, and the undecided label would not spread from the attacker to the attacked argument.

There are plenty of reasoning patterns, both in formal contexts and informal routine situations, in which humans adopt a mechanism where the undecidedness cast by conflicting arguments is confined and not propagated to the other parts of the decision making process.

In the formal context of Defeasible Logic, the standard semantics is ambiguity blocking. In Defeasible Logic a literal is ambiguous if there is a chain of reasoning concluding a and another concluding $\neg a$ and the superiority relation cannot resolve such conflict. The ambiguity blocking mechanism of defeasible logic is that both a and $\neg a$ are refuted and therefore any argument b_i based on a or $\neg a$ is refuted. The effect is that other arguments that are in conflict with b_i might result provable, rather than being deemed ambiguous as well.

The notion of ambiguity, as defined in Defeasible Logic, does not map to undecidedness in abstract argumentation. Ambiguous literals are most likely to be represented in abstract argumentation with two rebuttal arguments, which would indeed generate undecided arguments in some semantics, but not in all of them. Moreover, paradoxical situations like a cycle of three arguments with unidirectional attacks also generate an undecided situation in abstract argumentation, but they are not possible in Defeasible Logic since the superiority relation is required to be acyclic. Even if we consider the meaning of ambiguos in the English language (that is *a situation open to multiple interpretations*), it is clearly not the same as being undecided. For instance, in a paradox there is no ambiguity, since there are no multiple potentially valid interpretations but rather all the interpretations are contradictory.

An undecidedness blocking mechanism is certainly the most appropriate in a legal dispute. In the majority of legal systems, evidence in a criminal case has to satisfy the standard of proof *beyond reasonable doubt*. If evidence versus an accused is not definitive or open to multiple interpretations, then evidence is void and the judge rules in favour of the accused. Note the *asymmetry* of the principle: only evidence against the accused needs to be definitive (beyond reasonable doubt).

A typical example is the situation of two testimonies, both of them accusing x but providing conflicting accounts. In a Dung-style argumentation graph, the situation can be modelled as two rebuttal arguments (the testimonies) attacking the presumption of innocence of x that is the default position valid unless the contrary is proven (see Fig. 1). Note how arguments A and B undercut the validity of C, but not vice versa: C does not attack back A and B. This is because C is an example of defeasible argument, a default position whose validity is subject to exceptions. C is valid unless there is (valid) evidence against it. C does not rebut A or B since C is not based on evidence but rather on a presumption that can be retracted in light of the evidence provided by B and A. In order to rebut A and B, another kind of argument has to be provided, such as a testimony in favour of Carol's innocence. None of the complete semantics accepts the innocence of x and preferred and stable semantics sceptically reject the innocence of x. In this situation, an undecidedness blocking semantics seems to be more appropriate, since in a legal context the judge will consider x innocent.

Undecidedness blocking patterns are not confined to legal courts. An example is *the benefit of the doubt*, a mechanism routinely used by humans. When we reason using the benefit of the doubt, we tend to believe something even if we are not certain about it. The benefit of doubt is deeply embedded in human relationships, often granted to a person based on trust [16].

The proposal for an undecidedness blocking abstract semantics starts from an analysis of how undecided arguments affect the status of attacked arguments. In SCC-void semantics, only attacks from *in*-labelled arguments always prevent the attacked arguments to be accepted, while attacks from undecided arguments could have no effect, blocking the undecided label to spread, contrary to what would happen under Dung's complete semantics. The notion of admissibility is relaxed and the notion of effective attack is stronger, being harder for an argument to be excluded from the set of accepted arguments.

The reminder of the paper is organized as follows. Section 2 describes the required background of abstract argumentation. Section 3 provides a SCC-recursive definition of the new proposed SCC-void semantics, while Section 4 focuses on a discussion of their properties. Section 5 contains related works to date and section 6 concludes the paper and presents future research directions.

2. Background: Argumentation semantics

Definition 1. An argumentation framework AF is a pair $\langle Ar, R\rangle$, where Ar is a non-empty finite set whose elements are called arguments and $R \subseteq Ar \times Ar$ is a binary relation, called the attack relation. If $(a, b) \in R$ we say that a attacks b. Two arguments a, b are rebuttals *iff* $(a, b) \in R \wedge (b, a) \in R$, that means they define a symmetric attack. An argument a is initial if it is not attacked by any arguments, including itself.

An argumentation semantics identifies a set of sets of arguments that can survive the conflicts encoded by the attack relation R. Dung's semantics require a set of acceptable arguments to be conflict-free (an argument and its attacker cannot be accepted at the same time) and admissible (the set of arguments defends itself from external attacks).

Definition 2. A set $Arg \subseteq Ar$ is *conflict-free* iff $\nexists a, b \in Arg$ so that $(a, b) \in R$

Definition 3. A set $Arg \subseteq Ar$ defends an argument $a \in Ar$ iff $\forall b \in Ar$ such that $(b, a) \in R$, $\exists c$ such that $(c, b) \in R$. The set of arguments defended by Arg is denoted $F(Arg)$. A conflict-free set Arg is admissible if $Arg \subseteq$ and it is complete if $Arg = F(Arg)$.

The labelling approach of [4] is employed in this paper, where a semantics assigns to each argument a label in, out or undec.

Definition 4. Let $AF = \langle Ar, R\rangle$. A labelling is a total function $\mathcal{L} : Ar \rightarrow \{in, out, undec\}$. We write $in(\mathcal{L})$ or $\{a \in Ar|\mathcal{L}(a) = in\}$ $out(\mathcal{L})$ for $\{a \in Ar|\mathcal{L}(a) = out\}$ $undec(\mathcal{L})$ for $\{a \in Ar|\mathcal{L}(a) = undec\}$

Definition 5. ([4]). Let $AF = \langle Ar, R\rangle$. A complete labelling is one such that for every $a \in Ar$ it holds that:

1. if a is labelled *in* then all its attackers are labelled *out*;
2. if a is labelled *out* then it has at least one attacker that is labelled *in*;
3. if a is labelled *undec* then it has at least one attacker labelled *undec* and it does not have an attacker that is labelled *in*.

Definition 6. Given $AF = \langle Ar, R\rangle$,

1. $\mathcal{L}$ is the grounded labelling iff $\mathcal{L}$ is a complete labelling where $undec(\mathcal{L})$ is *maximal* (w.r.t. set inclusion) among all complete labellings of AF
2. $\mathcal{L}$ is a preferred labelling iff $\mathcal{L}$ is a complete labelling where $in(\mathcal{L})$ is *maximal* (w.r.t. set inclusion) among all complete labellings of AF
3. $\mathcal{L}$ is a stable labelling iff $\mathcal{L}$ is a complete labelling where $undec(\mathcal{L}) = \emptyset$
4. $\mathcal{L}$ is a semi-stable labelling iff $\mathcal{L}$ is a complete labelling where $undec(\mathcal{L})$ is minimal (w.r.t. set inclusion) among all complete labellings of AF

The labelling and the extension-based approach are equivalent. Given an argumentation framework and a semantics s for each set of in-labelled arguments there is an extension generated by the same semantics s and vice versa (proof in [4]).

An argumentation framework $AF = \langle Ar, R\rangle$ identifies a directed graph. The following are some graph-based definitions needed for the rest of the paper.

Definition 7. A subgraph of a graph $G = \langle Ar, R\rangle$ is a graph $G_s = \langle S, R_S\rangle$ whose set of nodes S is included in Ar and $R_S = R \cap (S \times S)$

A subgraph contains a subset of nodes of the original graph and any link whose endpoints are both in S (note how this subgraph is usually called a vertex induced subgraph). Given an argumentation framework AF, the restriction of an argumentation framework to a set of nodes S is the framework identified by a vertex induced subgraph of AF identified by the nodes S.

Definition 8. If G is a graph, a *strongly connected graph* of G is a subgraph of G where, for each pair of nodes $a, b \in G$ there is at least one directed path from a to b and at least one directed path from b to a. A *strongly connected component* (SCC) of G is a maximal (with respect to set inclusion) strongly connected subgraph.

Given a graph $G = \langle A, R\rangle$, we consider the graph composed by the strongly connected components of G. This is the graph G_{scc}= < S_{scc}, R_{scc},> where S_{SCC} is the set of strongly connected components of G, and there is a link from the strongly connected component S_1 to S_2 if at least an element of S_1 attacks an element of S_2 in the graph G. Formally, $\forall S_1, S_2 \in S_{SCC}, R_{SCC}(S_1, S_2)$ iff $\exists a \in S_1, \exists b \in S_2$ such that $R(a, b)$. The graph G_{scc} is a directed acyclic graph. Therefore, it is possible to define a topological ordering of such graph. A topological ordering of a graph G is an ordering such that, $\forall a, b \in G$, if $R(a, b)$ then $a \succ b$. Given G and G_{scc} in order to simplify the discussion in the paper, the following shortcut notation is introduced: for $a, b \in G$ we say that $a \succ b$ if a belongs to $S_1 \in G_{scc}$, b belongs to $S_2 \in G_{scc}$ and $S_1 \succ S_2$. Therefore, if a and b are in the same SCC, none of the two arguments precedes the other.

2.1. SCC-recursiveness and complete semantics

Definition 9. A given argumentation semantics $\mathcal{S}em$ is SCC-recursive if and only if for any argumentation framework $AF = \langle A, R\rangle$ its extensions are given by $E_{\mathcal{S}em}(AF) = \boldsymbol{GF}(AF, A)$, where $\boldsymbol{GF}(AF, A) \subseteq 2^A$ is defined as follows:

$\forall E \subseteq A$, $E \in \boldsymbol{GF}(AF, A)$ if and only if:

1. in case $|SCCS_{AF}| = 1$, $E \in \boldsymbol{BF}(AF, A)$
2. otherwise, $\forall S \in SCCS_{AF}$ is $(E \cap S) \in \boldsymbol{GF}\left(AF_{\downarrow UP_{AF}(S,E)}, \mathrm{U_{AF}\,(S, E) \cap C}\right)$

where $SCCS_{AF}$ is the set of the strongly connected components of AF, $\boldsymbol{BF}(AF, C)$ is a function, called base function, that, given an argumentation framework $AF = \langle A, R\rangle$, such that $|SCCS_{AF}| = 1$ and a set $C \subseteq A$ it returns a subset of 2^A. The set $UP_{AF}(S, E)$ is the set of all the arguments in S not attacked by the arguments already in the extension E, while $\mathrm{U_{AF}\,(S, E)} \subseteq UP_{AF}(S, E)$ is the set of arguments of S that are not attacked by an argument in the extension E and also defended by arguments in E.

Intuitively, the idea is that a semantics is computed by recursively analysing the strongly connected components of the argumentation graph [2]. At the beginning, the procedure is applied to the entire argumentation graph. If the graph is composed by more than one SCC, then it is decomposed in its SCCs and the extension is recursively built by analysing each SCC following the topological ordering of the acyclic graph identified by its SCCs. Portions of the graph that are represented by a single strongly connected component are analysed using a base function $\boldsymbol{BF}$ specific to each semantics. Therefore, the extensions of an initial SCC are identified using the base function $\boldsymbol{BF}$. A non-initial strongly connected component S is analyzed once all the SCCs preceding it in the topological ordering have been analysed. The extension of a non-initial SCC S is computed recursively by the function $\boldsymbol{GF}$ on a restriction of S, containing all the arguments in S except the ones attacked by those arguments already accepted (that are the arguments already in the extension E) and belonging to one of the SCCs previously analysed. This is the reason why the recursive step is applied to a restriction of S including the arguments in the set UP_{AF}, representing all the arguments in S not attacked by the arguments already in the extension E. In building the extension of S, the definition also considers the set of arguments in S that are attacked from outside by those arguments not in the extension E, but that in turn are defended by at least an argument in E (the set $U_{AF} \subseteq UP_{AF}$, the second parameter of the function $\boldsymbol{GF}$).

All Dung's complete semantics are SCC-recursive. We describe the computation of any complete Dung's semantics with the following variation of the SCC schema, which uses a different notation more convenient for our labelling-based discussion. For each complete semantics x, the base function is a function $\mathcal{L}_x$ turning the labelling of a graph consisting of a single SCC according to the semantics x. Following the general SCC-recursiveness schema, a SCC S is labelled by considering arguments in S but also the effect of the external attacks from arguments in SCCs preceding S in the topological ordering, that were labelled before S and independently from the labelling of S. In detail, some arguments in S could be attacked by arguments that have been labelled *out*, but these attacks are irrelevant in a complete labelling. Some arguments in S could be attacked by arguments labelled *in* (we refer to the arguments in S, attacked by *in*-labelled arguments, as the set Att_{in}) and some could be attacked by arguments labelled *undec*. We refer to the arguments in S attacked by *undec*-labelled arguments as the set Att_{undec}. The labelling of S is performed by applying $\mathcal{L}_x$ over the arguments of S after having labelled *out* the arguments in the set Att_{in} (and therefore only the restriction of S to $S \setminus Att_{in}$ is *de facto* labelled), and by considering that arguments in Att_{undec} are attacked by undecided arguments. Therefore, these arguments are labelled undecided if they are not labelled *out* (again, according to definition 5), and their undecided label might spread over other nodes in S.

Referring to the original SCC-recursiveness paper [2], we note how $S \setminus Att_{in} = UP_{AF}$. The set Att_{undec} carries the same information that in [2] is represented by the set U_{AF} (part of the second argument of $\boldsymbol{GF}$). Indeed, Att_{undec} represents what in the original SCC-recursiveness paper [2] is the set P_{AF} (*provisionally defeated* arguments), consisting of arguments of S attacked by arguments not in the extension (not labelled *in*) and not defended by an argument in the extension. Since $UP_{AF} = U_{AF} \cup P_{AF}$, then the two sets U_{AF} and $P_{AF} = Att_{undec}$ carry the same information in complementary ways. While U_{AF} represents the set of arguments of $UP_{AF} \subseteq S$ that could be part of the extension E, Att_{undec} is the set of arguments of $UP_{AF} \subseteq S$ that (even if not defeated by *in*-labelled arguments) cannot belong to the extension of S.

3. SCC-void undecidedness blocking semantics

Our discussion starts from the *beyond reasonable doubt* principle. This principle should protect arguments from doubtful attacks and make it harder to remove an argument from the set of accepted arguments. In the context of abstract argumentation, we propose to consider, on one hand, only the attacks from the accepted arguments always effective in making the attacked argument non-acceptable. On the other hand, the attacks coming from undecided arguments are *doubtful* and, under some circumstances, they have no effect. As a result, the undecided label is blocked and it is not assigned to the attacked argument.

In our proposal, an attack from an undecided argument a to an argument b is blocked (i.e. is ineffective) if a is in a SCC preceding the SCC containing the argument b in the topological order of the SCC graph. The rationale is the following: if two arguments are in the same SCC, each of them might affect the acceptability status of the other. Therefore, an argument could be undecided because of its own attacks to the other argument and vice versa. Both of the two arguments might be equally considered *not beyond reasonable doubt* and their attacks to each other considered ineffective. This could lead to accept both of them and potentially violate conflict-freeness[1]. Therefore our proposal is to defend an argument a from an undecided attacker b only if a is not involved in the assignment of the undecided label to its attacker b, condition that is guaranteed by the fact that the SCC containing b precedes in the topological order of the graph the SCC component containing a. This proposal blocks the propagation of *undecidedness* and limits it inside the strong connected component where it was generated.

Let us consider the graph in Fig. 2 that has two strongly connected components, one containing the arguments a and b and the other containing argument c only. Arguments a and b are forming a cycle and they are responsible for generating the undecided situation, while the label *undec* is propagated to the argument c that has no involvement in the cycle. Our proposal would be to keep a and b undecided, but accept c: the *undec* label is not propagated outside the cycle. In the floating assignment example (Fig. 1), our proposal implies that both a and b remain undecided, but c is accepted.

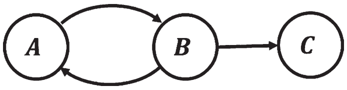

Fig. 2. Propagation of the undec label.

This is not the only possible way of modelling *undecidedness blocking*, but we believe it is one of the possible valid proposals. We formally define our new semantics, called SCC-void semantics, by modifying the SCC-recursive schema of complete semantics. For each Dung's complete semantics x, we define a corresponding SCC-void semantics, that is the semantics x where the above *undecidedness blocking* criterion is applied.

The SCC-void labelling for a complete semantics x, called $\mathcal{L}_{SCC_x}$, is a SCC-recursive labelling computed following the topological ordering of the strongly connected components of an argumentation graph. Arguments belonging to an initial strongly connected component S_i are labelled using the labelling function $\mathcal{L}_x$ of the chosen complete semantics x. We then apply our criterion: attacks from undecided arguments in S_i to arguments not in S_i are considered *not beyond reasonable doubt* attacks and thus they have no effect. These attacks are therefore discarded in the labelling of the subsequent SCCs in the topological order of the graph. As a result, the *undec* label is not propagated outside the SCC component where it was generated. Recursively, the arguments that are in a non-initial SCC S are labelled in the following way. First, we label *out* all the arguments in S included in the set Att_{in}, that are the arguments in S attacked by an *in*-labelled argument not belonging to S that was labelled in a previous step of the recursion. Then, we apply again the recursive step on the argumentation framework composed by the remaining arguments. We continue until all the arguments have been labelled. Therefore, while labelling a SCC S, we ignore the attacks coming from undecided arguments preceding S.

Note how the recursive scheme of the chosen complete semantics described in section 2.1 is followed, but we are not considering the external attacks from the set of undecided arguments when labelling an SCC. Using the terminology of the original SCC-recursiveness paper [2], neglecting attacks from external undecided arguments is equivalent to assume that the set $P_{AF} = \emptyset$ and therefore $UP_{AF} = U_{AF}$. We can therefore give the following two

[1] Unless we consider different labellings, where one of the two arguments is accepted and the other is not, generating preferred-semantics-like labellings.

extension-based and labelling-based SCC-recursive definitions of SCC-void semantics.

Definition 10. Given an argumentation framework $AF = \langle Ar, R\rangle$, E is an extension of the SCC-void semantics S_{SSC_x} if and only if:

1. in case $|SCCS_{AF}| = 1$, $E \in S_x(AF)$
2. otherwise, $\forall S \in SCCS_{AF}$, it is $(E \cap S) \in S_{SSC_x}\left(AF_{\downarrow U_{AF}(S,E)}\right)$

where $SCCS_{AF}$ is the set of strongly connected components of AF, $\mathbf{S_x}(AF)$ returns the extensions of a complete semantics x of an argumentation framework AF, and U_{AF} is the set of arguments in $S \subseteq Ar$ that are not externally attacked by an argument in the extension E. A labelling-based definition is also provided.

Definition 11. Let us consider the argumentation framework $AF = \langle A, R\rangle$, and $\mathcal{L}_x$, the labelling function for one of the complete semantics x. The SCC-void labelling for semantics x is identified by the function $\mathcal{L}_{SCC_x}$ idefined as follows:

1. in case $|SCCS_{AF}| = 1$, $\mathcal{L}_{SCC_x} = \mathcal{L}_x$
2. otherwise, $\forall S \in SCCS_{AF}$, it is

$$\mathcal{L}_{SCC_x} = \begin{cases} \mathcal{L}_{SCC_x} & \forall a \in S \setminus Att_{in}(S) \\ out & \forall a \in Att_{in}(S) \end{cases}$$

where $Att_{in}(S)$ is the set of arguments in S externally attacked by an in labelled argument: $Att_{in}(S) = \left\{a \in S | \exists b \in in\left(\mathcal{L}_{SCC_x}\right) : b \notin S \wedge R(b,a)\right\}$.

We note how the stable SCC-void semantics always coincides with Dung's complete stable semantics, since in a stable labelling there are no undecided arguments and therefore the undecidedness blocking criterion can never be applied.

3.1. Examples

In the *floating assignment* example (Fig. 3, part 1) using grounded SCC-void semantics, a and b are in an initial SCC and are therefore labelled *undec* using grounded semantics. However, their attacks are not propagated to c that is labelled *in*. The labellings of the preferred SCC-void semantics coincide with Dung's preferred semantics labellings, since in this case there are no undecided arguments.

In Fig. 3 (part 2), a is labelled *undec* and b is *in* for all the SCC-void semantics. In Fig. 3 (part 3), all the arguments are undecided in the grounded SCC-void labelling since, even if the attack from b to d could be neglected, the arguments d and e are still in a cycle. Therefore the grounded SCC-void labelling agrees with Dung's grounded labelling. For preferred SCC-void semantics there are two labellings: a, b, c are always undecided but, since the attack from b to d could be neglected, d is accepted and e is rejected in one labelling, while e is accepted and d is rejected in the other. Note that only the second labelling is a Dung's valid preferred labelling.

In Fig. 3 (part 4), by using the grounded SCC-void semantics arguments a and b are labelled *undec*, the initial argument e is *in*, d is *out* (defeated by e) and since we neglect the attack from b to c also c is accepted. Regarding preferred semantics, the two SCC-void preferred semantics labellings are also Dung's preferred labellings.

4. Discussion and properties

Table 1 (next page) shows the properties satisfied by SCC-void semantics and the corresponding Dung's complete semantics. As it is possible to note, the stable SCC-void semantics always coincides with Dung' stable semantics, since stable labelings do not contain undecided arguments.

SCC-void semantics are clearly non-admissible, since they can accept arguments that are not defended by *in*-labelled arguments. These semantics could be seen as employing a different form of admissibility since they still require an argument to be defended from the attacks of in-labelled arguments, but, under some conditions, not from the attacks of undec-labelled arguments.

All SCC-void semantics satisfy the reinstatement property since, if an argument has all its attackers labelled *out*, it is labelled *in*. However, SCC-void reinstatement is easier, since an argument a defeated by b is fully reinstated even by an argument c rebutting b, since c's attack makes b undecided and therefore the attack of b on a is not effective anymore. Our semantics are the only non-admissible semantics known to the authors satisfying reinstatement. Other non-admissible semantics, such as Stage semantics, allow an initial argument to be excluded from the extension, while both CF1 and CF2 semantics allow an argument whose attackers are all labelled *out* to be labelled *out*.

SCC-void semantics are also conflict-free and they satisfy rejection (*in*-labelled arguments always defeat attacked ones). The grounded SCC-void semantics

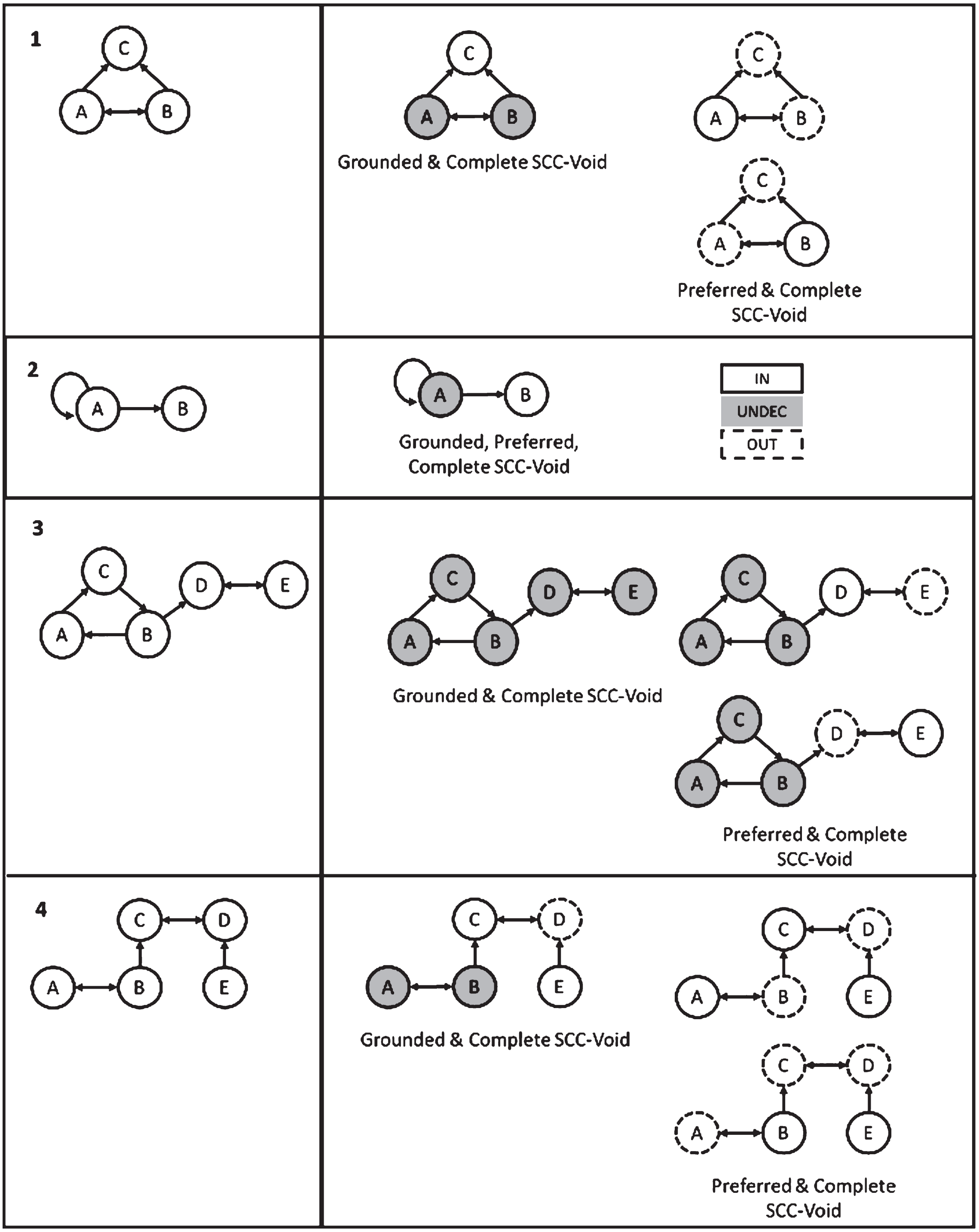
1
C
A
B
Grounded & Complete SCC-Void
Preferred & Complete SCC-Void
2
IN
UNDEC
OUT
Grounded, Preferred, Complete SCC-Void
3
D
E
Grounded & Complete SCC-Void
Preferred & Complete SCC-Void
4
Grounded & Complete SCC-Void
Preferred & Complete SCC-Void

Fig. 3. SCC-void labellings of argumentation frameworks.

always exists and it is unique, while the complete and preferred versions are multiple-status semantics with cardinality greater or equal than 1. Regarding directionality, all the SCC-void semantics (except the stable one) satisfy it, since the label assigned to an argument *a* does not depend on the label of the

Table 1
Properties of SCC-void semantics and complete semantics

Property	$\mathcal{L}_{SCC_{gr}}$	$\mathcal{L}_{gr}$	$\mathcal{L}_{SCC_{pr}}$	$\mathcal{L}_{pr}$	$\mathcal{L}_{SCC_{co}}$	$\mathcal{L}_{co}$	$\mathcal{L}_{SCC_{st}}$	$\mathcal{L}_{st}$
Admissible	N	Y	N	Y	No	Y	Y	
Conflict-free	Y	Y	Y	Y	Y	Y	Y	Y
Reinstatement	Y	Y	Y	Y	Y	Y	Y	Y
Rejection	Y	Y	Y	Y	Y	Y	Y	Y
Directionality	Y	Y	Y	Y	Y	Y	N	N
Abstention	N	N	N	N	N	Y	N	N
Cardinality	1	1	≥1	≥1	≥1	≥1	≥0	≥0
I-maximality	Y	Y	Y	Y	N	N	Y	Y

arguments following a in the topological order of the graph.

The property abstention states that, if an argument a is labelled *out* in at least one valid labelling and labelled *in* in at least another, then there must be a valid labelling where a is labelled *undec*. The grounded, preferred and stable SCC-void do not satisfy it, since Dung's grounded, preferred and stable semantics do not. Complete semantics satisfies it, but the complete SCC-void semantics does not. An example is in Fig. 3 (part 4) for arguments e and d. There is at least one complete SCC-void labelling with e accepted and d reject and one with d accepted and e rejected, but there is no complete SCC-void labelling where e or d are undecided.

A semantics satisfies I-maximality if no extension is a strict subset of another. It is not satisfied by Dung's complete semantics and therefore neither by the complete SCC-void semantics. It is indeed satisfied by the other SCC-void semantics since it is satisfied by the corresponding Dung's semantics.

4.1. *Relation between SCC-void acceptability and Dung acceptability*

It is interesting to study the relation between the set of *in*-labelled arguments of semantics x and the set of *in*-labelled arguments of the corresponding SCC-void semantics S_{SCC_x}.

In the SCC-void semantics, some of the arguments previously labelled *undec* could be promoted to the label *in*, and those arguments are now free to effectively attack other arguments. We wonder if some of the arguments accepted by x are now discarded by the corresponding SCC-void semantics. The following holds.

Theorem 1. If an argument a is at least *credulously* accepted by semantics x, then a is at least *credulously* accepted by SCC-void semantics for semantics x.

Proof. We prove that, if an argument a is labelled *in* in a complete labelling l, there is also a SCC-void labelling l_{scc} where a is labelled *in*. We first notice that arguments labelled *in,* within a complete labelling l, are indifferent to the undecided arguments. They either are initial arguments or defended by some *in*-labelled arguments (potentially including themselves). The same is for arguments labelled *out* in l: their label is assigned by the presence of an *in*-labelled argument. Moreover, in a complete labelling l, *in*-arguments do not receive any attacks from undecided arguments, and *undec* arguments only attack arguments labelled *undec* or *out*. In a SCC-void labellings, attacks from a subset of undecided arguments are neglected. Following this, two cases are possible:

Case 1. The neglected attacks are directed to *undec*-labelled arguments. In this case, the attacked arguments could be promoted to the label *in*. However, each new *in*-labelled argument b does not attack any *in*-labelled argument in l, but only arguments labelled *undec* and *out*, since b was undecided in l. Therefore the only potential effect of the attacks from b is that those arguments are now labelled *out* in l_{scc}. The effect of these new *out*-labelled arguments is potentially to change the label of some arguments previously undecided to *in*, and therefore $in\,(l) \subseteq in\,(l_{scc})$.

Case 2. The neglected attacks are directed to arguments labelled *out* in l. In this case, the effect is that each attacked argument c remains labelled *out* also in l_{scc}, since the *out* label of c in l was necessarily the effect of the attack from an *in*-labelled argument. This *in*-labelled argument is still labelled *in* in l_{scc}, since it cannot be affected by attacks included in case 1 above, and it is not affected by attacks included in case 2, since all the attacked arguments c remain labelled *out*, and therefore they do not affect any *in*-labelled arguments in l, and $in\,(l) \subseteq in\,(l_{scc})$. □

In case of a single-status semantics, like the grounded semantics, Theorem 1 means that the grounded SCC-void semantics is unique, always existing, and its extension is a superset of the Dung's grounded semantics. In particular, some undecided arguments could be promoted to *in* or demoted to *out*, while arguments labelled *out* and *in* under Dung's grounded semantics, retain their label in the grounded SCC-void labelling.

4.2. An alternative definition of SCC-void semantics

We defined SCC-void semantics as a modification of an underlying Dung's complete semantics x. By doing so, we have retained some desirable properties of Dung's semantics and a relation between the extensions prescribed by a semantics x and its SCC-void version. However, an alternative path could have been followed. In the literature, grounded, preferred and stable semantics are often defined as complete labellings whose set of *undec*- or *in*-labelled arguments satisfy some properties of maximality or minimality. For instance, a complete labelling is grounded iff it maximizes the set of *undec* arguments. We could have defined each SCC-void semantics in the same way, starting from the set of all complete SCC-void labellings.

However, the question is whether the two proposals coincide. Trivially, they coincide for stable SCC-semantics, since this semantics is equivalent to Dung's stable semantics. A weaker result holds for grounded and preferred, but the answer is negative.

Lemma 1. The grounded (respectively a preferred) SCC-void labelling maximizes the set of undecided arguments, (respectively maximizes the set of in-labelled arguments) among all the complete SCC-void labellings.

Proof. The proof is for grounded semantics, but the proof for preferred semantics is analogous. We have to prove that $undec\left(\mathcal{L}_{scc_{gr}}\right)$ is not contained in the *undec* set of any other SCC-void complete labellings $\mathcal{L}_{scc_x}$. The $\mathcal{L}_{scc_{gr}}$ labelling uses the grounded labelling as base function. Because of this, we can be certain that in all the initial SCCs the set of *undec* arguments is maximal. If in another complete SCC-void labelling $\mathcal{L}_{scc_x}$ there is at least one initial SCC S with a different labelling, then $undec\left(\mathcal{L}_{scc_x}(S)\right) \subset undec\left(\mathcal{L}_{scc_{gr}}(S)\right)$, and we prove lemma 1. If this is not the case, we then move to the next non-initial SCC S in the topological order. So far, the labels of all the arguments considered coincide with the labels assigned by the grounded SCC-void labelling $\mathcal{L}_{scc_{gr}}$. Therefore, every SCC-void labelling labels the same restriction of S in the next recursive step. The grounded semantics guarantees again that the labelling of S maximizes the set of undecided arguments. Therefore, in every SCC-void labelling $\mathcal{L}_{scc_x}$ generating a different labelling for S it is $undec\left(\mathcal{L}_{scc_x}(S)\right) \subset undec\left(\mathcal{L}_{scc_{gr}}(S)\right)$, and we prove the lemma. When all the arguments have been labelled, either all the complete SCC-void labellings agree with the unique grounded SCC-void labelling, or for every different SCC-void labellings $\mathcal{L}_{scc_x}$ there was at least one SCC S labelled in a different way for which $undec\left(\mathcal{L}_{scc_x}(S)\right) \subset undec\left(\mathcal{L}_{scc_{gr}}(S)\right)$ and we prove the lemma. □

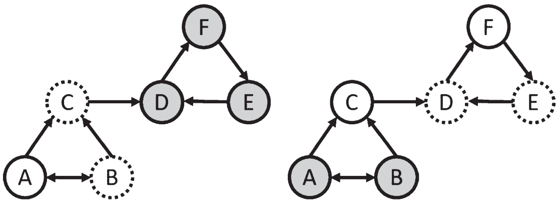

Fig. 4. A preferred SCC-void labelling (left) and the grounded SCC-void labelling for the same argumentation graph, both maximizing the set of *undec* arguments.

The above lemma is not a double implication. Not all the SCC-void labellings maximizing the *undec* set are grounded SCC-void labellings, and not all the labelling maximizing the set of *in*-labelled arguments are preferred SCC-void labellings.

For instance, labelling generated by $\mathcal{L}_{scc_{pr}}$ can also maximize the set of *undec* arguments. An example is given in Fig. 4, where both the grounded and the preferred SCC-void labellings maximize the set of *undec* arguments. The example also shows that the SCC-void labelling maximizing the set of *undec* arguments is not always unique and therefore a grounded SCC-void labelling defined as the labelling maximizing the number of undecided arguments would have lost that property. The example also shows that a labelling produced by $\mathcal{L}_{scc_{gr}}$ can also maximise the set of *in* labelled arguments.

4.3. Undecidedness Blocking in SCC-void semantics

SCC-void semantics are *undecidedness* blocking since, under some conditions, the label *undec* is not allowed to be transferred from the attacking to attacked arguments. Specifically, we can prove the following lemma, a straightforward consequence of our definition of SCC-void semantics:

Lemma 2 (Topological order sufficient condition). If all the undecided attackers of argument b belong to SCCs preceding the SCC containing b in the topological order of the graph, then b is labelled *in* or *out* in all the SCC-void labellings.

Proof. The lemma follows from our definition of SCC-void semantics. If all the undecided attackers of argument *b* belong to some SCCs preceding the one containing *b* in the graph, those attacks are ineffective, and therefore, since there are no more attacks to *b* from undecided attackers, argument *b* is either labelled *in* or *out*. □

Corollary. If *b* is not part of a cycle, than *b* is labelled *in* or *out* in all SCC-void labelligs.

The corollary implies that acyclic parts of the graph do not have undecided arguments, and the undecided label is not propagated outside the SCC where it was generated.

4.4. *Mixed decisions*

An agent might require different standards of proof for different arguments in the same discussion. For some arguments, it might require a high standard of proof such as *beyond reasonable doubt* (therefore blocking undecidedness), while for others it might require a weaker standard of proof. It indeed depends on the context, the attitude of the agent, how critical the decision is with respect to costs, benefits, risks and so forth. This implies that there could be labellings mixing Dung's complete semantics and SCC-void semantics, where in different parts of the graph different standards of proof are applied. For instance, let us consider the following situation.

Ann is the manager of Bob. Bob came late to work today and his justification was doubtful. Ann could grant the benefit of the doubt to Bob based on the fact that she knows him well and trust him, but the HR department could apply stricter rules and consider the doubtful justification not enough to avoid a penalty.

Figure 5 shows a situation in which the same argumentation graph is labelled by three different agents. The vertical bar on the attack shows where *undecidedness blocking* is used. Let us discuss the three situations:

(1) The agent uses Dung's grounded semantics and therefore it lets the undecidedness spread to all the arguments;

(2) The agent uses the grounded SCC-void semantics. The arguments in the cycle remain undecided, but argument *d* is now accepted since the attack from *b* is considered *not beyond reasonable doubt*;

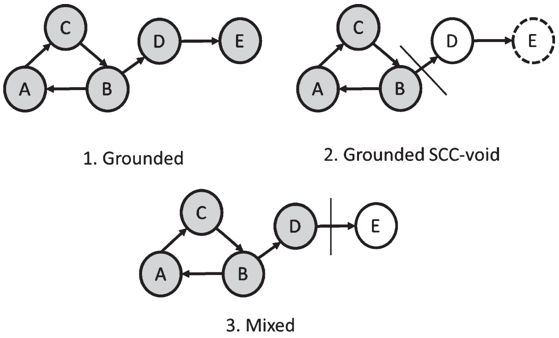

Fig. 5. A grounded, grounded SCC-void and a mixed labelling of the same argumentation graph. In the mixed labelling, the standard of proof *beyond reasonable doubt* is applied on the attack from *d* to *e*, but not on the attack from *b* to *d*.

(3) The agent uses a mixed strategy. It does not apply the standard of proof *beyond reasonable doubt* to defend argument *d* from the attack of argument *b*, but it applies the standard of proof to defend argument *e* from *d*. This could be, for instance, because argument *e* requires strict indisputable evidence to be rejected, while *d* does not.

5. Related works

Despite not coinciding with undecidedness blocking, the problem of ambiguity blocking and propagation in Defeasible logic (abbreviated DL) is certainly a relevant field of research. In defeasible logics, the problem of ambiguity blocking and propagation has been extensively studied. In particular, the work by [14] is the most relevant, since the authors propose an ambiguity propagating defeasible logics, and they provide a Dung-like argumentative version of both the standard (ambiguity blocking) and their ambiguity propagating semantics. Authors do not start from a modification of Dung's abstract semantics, but they propose a rule-based model of arguments using DL rules, and they study such structured argumentation systems using Dung's notion of acceptability. They show how their DL ambiguity propagating semantics can be obtained from Dung's grounded semantics postulates. We wonder how similar their ambiguity blocking semantics is to ours. As their semantics is for a structured rule-based argumentation system, we need to map some of their concepts to the abstract framework. Their semantics is a two-state (accepted/rejected) semantics. There are indeed arguments that cannot be labelled neither accepted or rejected, but they do not correspond to the notion of undecided arguments. De facto, they have no effect on other arguments, as they are rejected. Arguments built using ambiguous literals are marked as rejected. It is indeed easier for an argument to be accepted, since ambiguous literals that might prevent those argument to be accepted are rejected. The absence of a notion of undecidedness implies that, in Dung's term, the semantics would lose the reinstatement property, since a pair of otherwise non-attacked rebuttal arguments are both labelled out. The authors show how grounded semantics accepts less arguments than the standard DL-based semantics, in accordance with our Theorem 1, but it also rejects more than the standard DL-based semantics, in disagreement with our Theorem 1. This is due to the absence of an undecided status.

The principles of *beyond reasonable doubt* and of standard of proof have been extensively study in argumentation theory [13], but only few studies are relevant to abstract argumentation. In the context of structured argumentation, we mention the work of Prakken [20] on modelling standards of proof, and the modification of the Carneades framework [12].

Regarding abstract argumentation, the most explicit study about standard of proof is [1]. Here, the authors consider how each Dung's semantics has a different level of cautiousness that is mapped to a corresponding legal standard of proof. Only initial arguments are beyond doubt, but they consider the sceptically preferred justification a beyond reasonable doubt position. In the floating assignment example (Fig. 1), the authors recognize the two attackers as doubtful, but they consider the sceptically preferred rejection of c beyond reasonable doubt. It could be noticed that this position is failing to acknowledge that, if each of the attackers are considered doubtful, their effect cannot be (at last in all the situations) *beyond doubt*.

Brewka et al. [3] also criticises [1] since they doubt the fact that various Dung's semantics can capture the intuitive meaning of legal standard of proof (detailed discussion in here [13]). In the case of beyond reasonable doubt, we agree with Brewka, complete Dung's semantics are not adequate to model this principle. Prakken has analysed the floating assignment and its link to standard of proof in his work [21], where he responds to objections advanced by Horty in [15]. Prakken underlines that, in many problematic situations, including the floating assignment, there could be hidden assumptions about the specific problem which, if made explicit, are nothing but extra information that defeat the defeasible inference. In the case of the floating assignment, Prakken agrees that if beyond reasonable doubt is our standard of proof – like in a criminal case where there are two conflicting testimonies - we should not conclude that the accused is guilty. However, this does not mean that argumentation semantics are somehow invalid. In the case of conflicting testimonies, as already showed by Pollock [19], the situation could be correctly modelled by making some hidden assumptions explicit and adding extra arguments to model such assumptions. In the conflicting testimonies, the fact that two witnesses contradict each other is a reason to add an argument undercutting the credibility of both. However, the problem of when to add arguments and how they interact with existing arguments has still to be faced, and in this work we have tackled it by embedding assumptions in an abstract argumentation semantics rather than adding arguments.

In his presentation of semi-stable semantics, Caminada [5] also clarifies the logical assumptions beyond the treatment of the floating assignment. In particular, he observes how the preferred semantics solution is based on the assumption that we know with certainty that one of the two attacking arguments is valid, since in this case we do not need to know which one is valid in order to safely discard c.

6. Conclusions and future works

In this paper we have explored a new family of semantics where undecidedness is blocked. We proposed to consider attacks from undecided arguments a weaker form of attacks, considering them *not beyond reasonable doubt*. We have proposed to neglect attacks from undecided arguments under some conditions and we have defined the SCC-void semantics using an SCC-recursive schema and proved their fundamental properties.

For each complete semantics, it is possible to define a corresponding SCC-void semantics. SCC-void semantics are conflict-free, non-admissible (in Dung's sense), but employing a defence-based relaxed notion of admissibility, they allow reinstatement and they accept credulously what the corresponding complete semantics accepts at least credulously. SCC-void semantics block the propagation of undecidedness over an argumentation graph, by confining it to the SCC where the undecided situation was generated. We believe to have proposed a novel and well-motivated contribution to abstract argumentation semantics.

Future work will focus on blocking undecidedness in quantitative extensions of Dung's abstract frameworks such as probabilistic [8, 10], fuzzy [7, 9] or graded [6, 27] argumentation frameworks. Moreover, we intend to study the application of undecidedness blocking semantics in contexts such as the medical field [17, 25] or persuasive dialogues [26].

References

[1] K. Atkinson and T. Bench-Capon, Argumentation and standards of proof, *11th international conference on Artificial intelligence and law*, (2007), 107–116. ACM.

[2] P. Baroni, M. Giacomin and G. Guida, Scc-recursiveness: A general schema for argumentation semantics, *Artificial Intelligence* **168**(1-2) (2005), 162–210.

[3] G. Brewka and T.F. Gordon, Carneades and abstract dialectical frameworks: A reconstruction. In *Proceedings of the 2010 conference on Computational Models of Argument: Proceedings of COMMA 2010* (2010), 3–12. IOS Press.

[4] M.W. Caminada and D.M. Gabbay, A logical account of formal argumentation, *Studia Logica* **93**(2) (2009), 109–145.

[5] M.W. Caminada, W.A. Carnielli and P.E. Dunne. Semistable semantics, *Journal of Logic and Computation* **22**(5) (2012), 1207–1254.

[6] P. Dondio, Ranking Semantics Based on Subgraphs Analysis. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi Agent Systems* (2018), (pp. 1132-1140).

[7] P. Dondio, Propagating degrees of truth on an argumentation framework: An abstract account of fuzzy argumentation. In *Proceedings of the Symposium on Applied Computing* (2017). (pp. 995-1002). ACM.

[8] P. Dondio, Toward a computational analysis of probabilistic argumentation frameworks, *Cybernetics and Systems* **45**(3) (2014), 254–278.

[9] P. Dondio, Multi-Valued and Probabilistic Argumentation Frameworks. In *COMMA* (2014), pp. 253–260.

[10] H. Li, N. Oren and T.J. Norman, Probabilistic argumentation frameworks. *International Workshop on Theorie and Applications of Formal Argumentation. Springer, Berlin, Heidelberg*, 2011.

[11] 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), 321–357.

[12] T.F. Gordon and D. Walton, The carneades argumentation framework–using presumptions and exceptions to model critical questions. In *6th computational models of natural argument workshop (CMNA), European conference on artificial intelligence (ECAI), Italy* **6** (2006), 5–13.

[13] T.F. Gordon and D. Walton, Proof burdens and standards. In *Argumentation in artificial intelligence. Springer.* (2009), 239–258.

[14] G. Governatori, M.J. Maher, G. Antoniou and D. Billington, Argumentation semantics for defeasible logic, *Journal of Logic and Computation* **14**(5) (2004), 675–702.

[15] J.F. Horty, Argument construction and reinstatement in logics for defeasible reasoning, *Artificial Intelligence and Law* **9**(1) (2001), 1–28.

[16] A. Zaheer, B. McEvily and V. Perrone, Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance, *Organization Science* **9**(2) (1998), 141–159.

[17] L. Longo, B. Kane and L. Hederman, Argumentation theory in health care. In *Computer-Based Medical Systems (CBMS)* (2012), (pp. 1-6). IEEE.

[18] F. Maier and D. Nute, Ambiguity propagating defeasible logic and the well-founded semantics. In *European Workshop on Logics in Artificial Intelligence* (2006), pp. 306–318. Springer.

[19] J.L. Pollock, Cognitive carpentry: A blueprint for how to build a person. MTI Press. 1995.

[20] H. Prakken and G. Sartor, On modelling burdens and standards of proof in structured argumentation. In *JURIX* (2011), 83–92.

[21] H. Prakken, Intuitions and the modelling of defeasible reasoning: Some case studies. (2002), arXiv preprint cs/0207031.

[22] I. Rahwan and G.R. Simari, eds. Argumentation in artificial intelligence. Vol. 47. Heidelberg: Springer, 2009.

[23] P. Baroni and M. Giacomin, Some Considerations on Epistemic and Practical Reasoning in Abstract Argumentation. *Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence, 20-23 November 2018, Trento, Italy*. 2018.

[24] P. Baroni, M. Caminada and M. Giacomin, An introduction to argumentation semantics, *The Knowledge Engineering Review* **26**(4) (2011), 365–410.

[25] L. Longo and P. Dondio, Defeasible reasoning and argument-based systems in medical fields: An informal overview. In 2014 IEEE *27th International Symposium on Computer-Based Medical Systems* (2014), (pp. 376-381). IEEE.

[26] A. Hunter, Computational Persuasion with Applications in Behaviour Change. In *COMMA* (2016), (pp. 5-18).

[27] L. Amgoud and J. Ben-Naim, Ranking-based semantics for argumentation frameworks. *International Conference on Scalable Uncertainty Management. Springer, Berlin, Heidelberg*, 2013.

[28] P. Dondio, Multi-Valued and Probabilistic Argumentation Frameworks. In *COMMA* (2014), (pp. 253-260).

[29] D. Nute, Defeasible logic, *International Conference on Applications of Prolog. Springer, Berlin, Heidelberg*, 2001.