A Proposal to Embed the In Dubio Pro Reo Principle into Abstract Argumentation Semantics based on Topological Ordering and Undecidedness Propagation

Pierpaolo Dondio

School of Computer Science, Dublin Institute of Technology,
Kevin Street, Dublin 2, Ireland.
pierpaolo.dondio@dit.ie

Abstract. In this paper we discuss how the *in dubio pro reo* principle and the corresponding standard of proof *beyond reasonable doubt* can be modelled in abstract argumentation. The *in dubio pro reo* principle protects arguments against attacks from doubtful arguments. We identify doubtful arguments with a subset of undecided arguments, called active undecided arguments, consisting of cyclic arguments responsible for generating the undecided situation. We obtain the standard of proof *beyond reasonable doubt* by imposing that attacks from doubtful undecided arguments are not enough to change the acceptability status of an attacked argument (the *reo*). The resulting semantics, called SCC-void semantics, are defined using a SCC-recursive schema. The semantics are conflict-free, non-admissible (in Dung's sense), but employing a more relaxed defence-based notion of admissibility; they allow reinstatement and they accept credulously what the corresponding complete semantics accepts at least credulosly.

1 Introduction

Abstract argumentation is a framework for non-monotonic reasoning, where conclusions are reached by evaluating arguments and their conflict relations. The theory is centered on the notion of argumentation framework [10], a directed graph where nodes represent arguments and links represent an attack relation defined over arguments. One of the main tasks of abstract argumentation is the computation of the acceptability status of arguments. This task is performed by the application of an argumentation semantics, a set postulates used to identify the sets of arguments, called extensions, which successfully survived the conflicts encoded in the attack relation. In the labelling approach [4], the effect of an argumentation semantics is to assign to each argument a label *in, out* or *undec*, meaning that the argument is respectively accepted, rejected or deemed undecided. The *undec* label represents a situation in which the semantics has not enough reasons to accept or reject an argument.

In this paper we explore the definition of a new family of abstract semantics, called SCC-void semantics. For each complete semantics, it is possible to define a SCC-void version of it. SCC-void semantics are conflict-free and non-admissible semantics, but still employing 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.

The rationale behind the SCC-void semantics is the aim of modeling the *in dubio pro reo* principle into abstract argumentation. From one hand, we claim that the current abstract semantics do not embed the *in dubio pro reo* principle in their postulates, and on the other hand we aim to show how it is possible to model this principle using a purely abstract argumentation semantics. The sentence *in dubio pro reo* translates in English as "*when in doubt, in favour of the accused*" and it is a legal principle according to which, if evidence against an accused A is not definitive but it leaves doubts about its validity or it leads to multiple interpretations, it should be ruled in favour of A (the reo). The principle not only implies the *presumption of innocence*, that sets the burden of proof on the prosecutor, but it also implies that the standard of proof needed to prove that an accused is guilty must be beyond reasonable doubt. In other words, evidence that are not conclusive are deemed void.

In terms of attacks and extensions, the *in dubio pro reo* principle should make harder for an argument to be excluded from an extension. Current complete semantics do not embed this principle. Under grounded semantics an attack from an undecided argument is enough to remove an argument from the extension, while preferred semantics allows attacks from credulously accepted arguments to make attacked arguments skeptically rejected.

This is exemplified by the well-known floating assignment example, shown in Figure 1. All the abstract argumentation semantics reported in literature either label c undecided or rejected. Grounded semantics, representing the most skeptical position, labels the three arguments undecided. Preferred semantics generates two symmetrical labellings, one with a accepted and b and c rejected, and one with b accepted and a and c rejected, so that c is skeptically rejected (since it is rejected in all the labellings). Stable and semi-stable semantics agree with the preferred semantics, as do the non-admissible Stage, CF1 and CF2 semantics. Only in the naive semantics accepting all the conflict-free subsets of a graph, argument c is accepted, but losing conflict-freeness is not desirable.

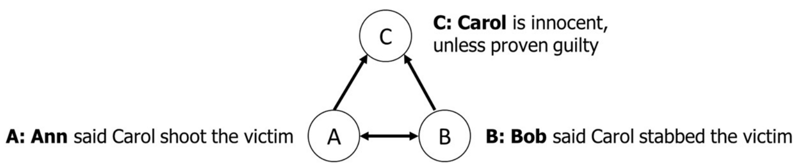

Fig. 1: The floating assignment in a legal context

If we instantiate the floating assignment in a criminal case, argument c represents the *presumption of innocence* of Carol, that is innocent unless proven guilty. It is therefore a defeasible argument that can be unidirectionally attacked if new evidence against her is provided. Arguments b and a could represent two equally reliable witnesses called Bob and Ann at a trial against Carol. Bob said that Carol is guilty since he saw Carol stabbing the victim. Ann also said that Carol is guilty since she saw her shooting the victim. The victim's body was never found. According to the *beyond reasonable doubt* principle, the two contradictory testimonies are void. There is no legal evidence against Carol and she is free to go. An abstract argumentation semantics based on the *in dubio pro reo* principle would therefore assign the label *in* to argument c. Regarding a and b, we believe they should be left undecided, to mark their conflicting status.

The fact that none of the current semantics models the *in dubio pro reo* principle does not count as a weakness. First, those semantics embed different assumptions in their postulates. There is no single correct treatment of the floating assignment example, since it depends on contextual information, standard of proof, criticality of the information, perceived level of disagreement between a and b and so on. The preferred semantics solution is totally valid if we know with certainty that either Ann or Bob are saying the truth, since in any interpretations Carol results guilty. The point is that there could be assumptions that are hidden in the analysis. Pollock [16] and Prakken [17-18] already suggested that the situation of inconsistent witnesses could be handled by adding two new arguments to the graph, each of them undercutting one of the witnesses, since the fact that a testimony a is rebutted by a testimony b is an argument for a to be discarded (and vice-versa). Far from claiming that current semantics are not adequate, our challenge is to define a well-behaved abstract argumentation semantics embedding the *in dubio pro reo* principle. Modelling it with the addition of new arguments leaves open the problem of when to add those arguments and how these additional arguments interact with existing arguments. Here we propose a family of semantics, called SCC-void semantics, that are based on a weaker notion of admissibility and that retain the majority of good properties of complete semantics. The starting idea is that the principle of *in dubio pro reo* is modelled in abstract argumentation by reconsidering the effect of undecided arguments. An attack from an undecided argument (the *doubtful* argument) should not change the acceptability status of the attacked argument (the reo). In an *in dubio pro reo*-labelling, only *in*-labelled arguments should be able to remove attacked arguments from the extension, while undecided arguments should not be strong enough to generate effective attacks. However, neglecting attacks from undecided arguments is a vague and potentially misleading statement. In the floating assignment example, using grounded semantics, one could say that all the three arguments should be turned into accepted arguments, on the basis that there is no conclusive evidence regarding their rejection. This would not only generate non-admissible semantics, but also semantics losing the property of conflict-freeness accepting sets of arguments internally attacking each other.

Our central idea is that the *in dubio pro reo* principle protects an argument a against the attacks of doubtful undecided arguments only if a is not involved in generating the undecided situation attacking a. We believe this is one of the possible faithful translations of the principle. In other words, we require the (doubtful) evidence against an accused to be external to the accused; the accused has no involvement in it. Note how the involvement of the accused would generate a situation of conflict of interest where evidence, independently from their reliability, could not even be considered as valid evidence.

Under complete semantics there are two distinct reasons why an argument is labelled *undec*. Referring to Figure 2a and grounded semantics, we note how arguments a

and b are responsible for the undecided situation by forming a cycle, and we refer to them as *active* undecided arguments. On the other hand, the label *undec* is propagated to *c*, that has nothing to do with the situation generating the undecided conflict, and it is therefore a *passive* undecided argument. Our proposal is that attacks from active undecided arguments have no effect on the status of other arguments (excluding themselves). This implies that some of the passive undecided arguments might be promoted to accepted arguments. In the paper, we discuss a way to partition undecided arguments into active and passive arguments based on the topological ordering of the strongly connected components of the argumentation graph.
Our proposal generates a new family of semantics, called SCC-void semantics. In the paper, their formal definition is provided using the SCC-recursiveness schema. For each Dung's complete semantics x, an SCC-void version can be defined. The semantics x is responsible for the identification of undecided arguments, on which our additional *in dubio pro reo* principle is applied to generate potentially new labellings. For instance, in case of the floating assignment, the grounded SCC-void semantics would start labelling a and b undecided, and it would then label c *in*, since the undecided labels of a and b are not propagated outside the SCC created by those arguments. The arguments a and b represent a doubtful situation that, according to the *in dubio pro reo* principle, should not affect the acceptability status of c.

The paper is organized as follows. The next section introduces the required background of abstract argumentation. Section 3 provides a SCC-recursive definition of our SCC-void semantics. Section 4 contains a discussion of the properties of our semantics. Section 5 contains related works before our conclusions.

2 Background: Abstract Argumentation Semantics

Definition 1. An argumentation framework AF is a pair $< Ar, R >$, 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* $(a, b) \in R \wedge (b, a) \in R$, i.e. 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 $(b, a) \in R, \exists c \in Arg$ 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 $Arg \subseteq F(Arg)$ and it is complete if $Arg = F(Arg)$.

We follow the labelling approach of [4], where a semantics assigns to each argument a label *in, out* or *undec*.

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

Definition 5. ([4]). Let $AF = < Ar, R >$. A complete labelling is a labelling such that for every $a \in Ar$ 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 $AF = < Ar, R >$,

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

An argumentation framework $AF = < Ar, R >$ identifies a directed graph. The following are some graph-based definitions needed for the rest of the discussion.

Definition 7. A subgraph of a graph $G = < Ar, R >$ is a graph $G_s = < S, R_S >$ 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 is a framework identified by a vertex induced subgraph of AF.

Definition 8. Let us consider an argumentation framework $AF = < Ar, R >$, and a set of nodes $S \subseteq Ar$. The restriction of AF to S, written $AF_{\downarrow S}$, is the argumentation graph $AF_{\downarrow S} = < S, R_S >$ where $R_S = R \cap (S \times S)$.

Definition 9. 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 (w.r.t. set inclusion) strongly connected subgraph.

Given a graph $G = < A, R >$, 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 is connected to 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 we introduce the following shortcut notation: 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 $S_1 \succ S_2$.

2.1 SCC-recursiveness and Complete Semantics

Definition 10. A given argumentation semantics S is SCC-recursive if and only if for any argumentation framework $AF = < A, R >$, its extensions are given by $E_S(AF) = \boldsymbol{GF}(AF, A)$, 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 $(E \cap S) \in \boldsymbol{GF}(AF_{\downarrow UP_{AF}(S,E)}, \mathrm{U_{AF}(S, E)} \cap \mathrm{C})$

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

The idea [2] is that a semantics is computed by recursively analysing the strongly connected components of the argumentation graph. At the beginning, the procedure is applied to the entire argumentation graph. If the graph is composed by more than one SCC, 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. Portion 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, extensions of an initial SCC are labelled using the base function $\boldsymbol{BF}$. A non-initial strongly connected component S is labelled once all the SCCs preceding it in the topological ordering have been analysed. The extension of a non-initial SCC S is computed by the base function on a restriction of S containing all the arguments in S minus the ones attacked by arguments already accepted (=already in the extension E) and belonging to SCCs previously analysed. This is 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 argument already in the extension E. In building the extension of S, the algorithm could also consider the set of arguments in S that are attacked from outside by 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 $\mathcal{L}_x$ returning the labellings of a graph consisting of a single SCC according to

semantics x. Following the general SCC-recursiveness schema, a SCC S is labelled by considering arguments in S but also the effect of external attacks from arguments in SCCs preceding S in the topological ordering, that can be labelled before S and independently from the labelling of S. In particular, some arguments in S could be attacked by arguments that have been labelled *out*, and therefore these attacks are irrelevant in a complete labelling. Some arguments in S could be attacked by arguments labelled *in* (we call the arguments in S attacked by *in*-labelled arguments the set Att_{in}) and some could be attacked by arguments not in the extension (not labelled *in*) but not defended by an argument in the extension (and therefore not labelled *out*). For definition 5, it follows that these attacking arguments must be labelled *undec* by a previous application of $\mathcal{L}_x$. We call the arguments in S attacked by *undec*-labelled arguments the set Att_{undec}.
The labelling of S is done 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 and therefore they are labelled undecided if they are not labelled *out* (again, for definition 5), and that their undecided label might spread over other nodes in S.
Note that we have followed the SCC-recursiveness schema using a different notation. Referring to the original SCC-recursiveness paper, $S \setminus Att_{in} = UP_{AF}$. The set Att_{undec} carries the same information that in the original SCC-recursive schema is represented by the set U_{AF} (part of the second argument of $\boldsymbol{GF}$), but Att_{undec} represents what in the original SCC-recursiveness paper 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 $UP_{AF} = U_{AF} \cup P_{AF}$, 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 Semantics

The *in dubio pro reo* principle should protect arguments against doubtful attacks. We propose to identify doubtful attacks with attacks coming from undecided arguments. However, as discussed in the introduction, we propose to apply the principle of the *in dubio pro reo* under a condition: the attacked argument a is not actively involved in generating the undecided situation attacking a. This was proposed mainly to avoid a situation of conflict of interest.

The first step is the identification of arguments *actively* involved in the undecided situation (called *active* undecided arguments) and arguments that are not involved, called *passive* undecided arguments. The latter are the ones that could benefit from the *in dubio pro reo* principle. Among the possible proposals, we suggest

to partition *passive* and *active* undecided arguments using the topological ordering of the graph.
Under complete semantics x, a label *undec* is assigned to an argument a because one or more of its attackers is labelled *undec*. A *passive* undecided argument a is an argument receiving the undecided label because of one or more attacks from undecided arguments preceding a in the topological ordering. If all those attacks were ignored, a would not be labelled undecided by semantics x anymore, meaning that a is not the cause of the undecided situation. On the contrary, an *active* undecided argument would still be labelled undecided, meaning that the argument is part of a cycle of undecided arguments and actively takes part in generating the undecided situation.

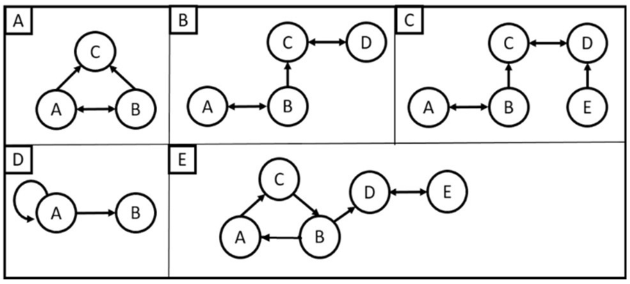

Figure 2. Argumentation Graphs discussed in the paper

We want to build a *pro reo* semantics where an argument a does not have to be defended from undecided arguments if it is not involved in generating the undecided situation. The *in dubio pro reo* principle joint to our topology-based criterion implies that arguments are protected from attacks coming from undecided arguments preceding them in the topological ordering.

We obtain such 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 our *in dubio pro reo* 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 the 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 the *pro reo* criterion: attacks from undecided arguments in S_i to arguments not in S_i are considered *not beyond reasonable doubt* and they have no effect. These attacks are therefore discarded in the labelling of subsequent SCCs in the topological ordering and the *undec* label is not propagated outside the SCC component where it was generated.
Recursively, 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 At_{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, that is a

restriction of the original argumentation graph. We continue to do so until all the arguments have been labelled. Therefore, when labelling a SCC S, we ignore the attacks coming from undecided arguments preceding S.
Note how we are following the recursive scheme of the chosen complete semantics described in section 3, with the only modification that we are not considering the external attacks from the set of undecided arguments when labelling a SCC. Using the terminology of the original SCC-recursiveness paper, neglecting attacks from external undecided arguments is equivalent to assume that the set $P_{AF} = Att_{undec} = \emptyset$ and therefore $UP_{AF} = U_{AF}$. We can therefore give the following extension-based SCC-recursive definition of SCC-void semantics.

Definition 12. Given an argumentation framework $AF =< Ar, R >$, E is an extension of the SCC-void semantics S_{SSC_x} if and only if:

1. in case $|SCC_{AF}| = 1$, $E \in \boldsymbol{S}_x(AF)$
2. otherwise, $\forall S \in SC_{AF}$, it is $(E \cap S) \in S_{SSC_x}(AF_{\downarrow U_{AF}(S,E)})$

where $\boldsymbol{S}_x(AF)$ 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.
The definition considers that $UP_{AF} = U_{AF}$, and therefore the function S_{SSC_x} needs only one parameter. We also provide a labelling-based definition.

Definition 13. Let us consider the argumentation framework $AF =< A, R >$, and the function $\mathcal{L}_x$ computing the labelling of a complete semantics x. The SCC-void labelling for semantics x is identified by the function $\mathcal{L}_{SC_x}$, defined as follows:

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

$$\mathcal{L}_{SCC_x}(S) = \begin{cases} \mathcal{L}_{SC_x}(AF_{\downarrow S \setminus Att_{in}(S)}), \forall a \in S \setminus Att_{in}(S) \\ out, \quad \forall a \in Att_{in}(S) \end{cases}$$

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

3.1 Examples

In the *floating assignment* example (Figure 2a) 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*. For preferred, semi-stable and stable SCC-void semantics, there are no undecided arguments, so the two valid labellings are also preferred, semi-stable and stable SCC-void labellings.
In Figure 2d, a is labelled *undec* and b is *in* for all the SCC-void semantics.
In Figure 2e, all the arguments are undecided in the grounded SCC-void labelling since, even if the attack from b to d is neglected, the arguments d and e are still in a cycle. Therefore the grounded SCC-void labelling agrees with the 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 is 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 Figure 2c, 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 preferred semantics labellings are also SCC-void preferred labellings.

4 DISCUSSION AND PROPERTIES

By relying on the properties of the underlying Dung's complete semantics x, we can prove these properties for S_{SCC_x}:

Lemma 1. If x is a complete semantics, the semantics S_{SCC_x} satisfies the following:

1. It is conflict free and non-admissible (in Dung's sense)
2. It allows reinstatement

SCC-void semantics are clearly non-admissible, since they can accept arguments not defended by in-labelled arguments, as in the floating assignment example. They could be seen as employing a different form of admissibility, weaker than Dung's admissibility. We still require an argument to be defended, but in a SCC-void semantics attacks from (some) undecided arguments are considered too weak and neglected, so an argument is not required to be defended from them. SCC-void semantics are still based on a concept of defence as the admissibility-based semantics, and they still satisfy the reinstatement property since, if an argument has all its attackers labelled out, it is labelled in. However, the *in dubio pro reo* principle makes reinstatement easier, since an argument a defeated by b (initial) is fully reinstated even by an argument c rebutting b, since the doubt casted by c's attack is enough to make the attack from b to a no more *beyond reasonable doubt*.

SCC-void semantics are the only non-admissible semantics known to the author 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.

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 $S_{SC\ x}$. Since attacks from active undecided arguments are neglected, 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, which would not be desirable. The following holds.

Theorem 1.
If an argument is at least *credulously* accepted by semantics x, it 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* in a complete labelling l are indifferent to 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. There are two cases:

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 some arguments labelled *undec* in l are now labelled *out* in l_{scc}, 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 $in(l) \subseteq in(l_{scc})$. □

By promoting some arguments to the *in* status, an SCC-void semantics might generate additional valid labellings affecting the justification status of arguments. For instance, if we consider the graph in Figure 2d, there is one valid preferred labelling where e is accepted, a, b, c and d undecided, that is also an SCC-void labelling. Therefore e is at least credulously accepted by the preferred SCC-void semantics. However, by neglecting the attack from b to d, the preferred SCC-void semantics generates also the labelling where d is accepted, a, b, c undecided and e rejected and therefore e and d are credulously accepted by the SCC-void semantics.
In case of a single-status semantics, like the grounded, theorem 1 means that the grounded SCC-void semantics is unique, always existing, and its extension is a superset of the complete 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 (Figure 3).

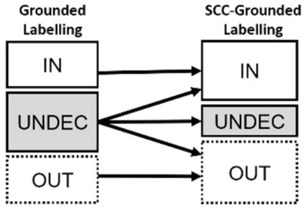

Fig. 3: $in(\mathcal{L}_{gr}) \subseteq in\left(\mathcal{L}_{scc_{gr}}\right)$,$out(\mathcal{L}_{gr}) \subseteq out\left(\mathcal{L}_{scc_{gr}}\right)$,$undec\left(\mathcal{L}_{scc_{gr}}\right) \subseteq undec(\mathcal{L}_{gr})$.

4.1 An alternative definition of SCC-void semantics

We defined each SCC-void semantics as a modification of an underlying Dung's complete semantics x, a less skeptical version of x embedding the *in dubio pro reo* principle in the way attacks are evaluated. 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 literature, grounded, preferred, stable and semi-stable semantics are often defined as complete labellings whose set of *undec*- or *in*-labelled arguments satisfy some properties of maximality or minimality. The grounded semantics is for instance the complete labelling maximizing the set of *undec* arguments. We could have defined the SCC-void semantics in the same way, starting from the set of complete SCC-void labellings. We first wonder if the two proposals coincide. A weaker result holds, but the answer is negative. It can be proved that:

Lemma 2. *A grounded (preferred/semistable/stable) SCC-void labelling is a complete SCC-void labelling maximizing the set of undecided arguments (maximizing the set of in-labelled arguments/minimizing the set of undecided arguments/where the set of undecided arguments is empty).*

Proof. The proof is for grounded semantics (the other proofs are 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 $\mathcal{L}_{sc\ gr}$ labelling uses the grounded labelling as base function. Because of this, we are sure 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 $undec(\mathcal{L}_{scc_x}(S)) \subset undec\left(\mathcal{L}_{scc_{gr}}(S)\right)$, and we prove the thesis. If this is not the case, we move to the next non-initial SCC S in the topological ordering. 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 $undec(\mathcal{L}_{scc_x}(S)) \subset undec\left(\mathcal{L}_{scc_{gr}}(S)\right)$, and we prove the thesis.
When all the arguments of 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 $undec(\mathcal{L}_{sc\ x}(S)) \subset undec\left(\mathcal{L}_{sc\ gr}(S)\right)$ and we prove the thesis. □

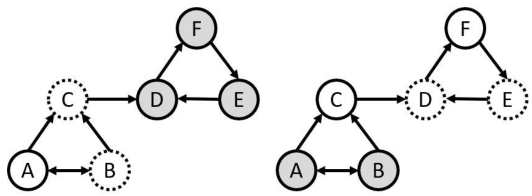

Fig. 4: A preferred SCC-void labelling (left) and the grounded SCC-void labelling for the same argumentation graph, both maximizing the set of *undec* arguments.

The above lemma is not a double implication, except for the stable semantics. 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. Labelling generated by $\mathcal{L}_{scc_{pr}}$ can also maximize the set of *undec* arguments. An example is given in Figure 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 labellng maximizing the set of *undec* arguments is not always unique and therefore a grounded SCC-void labelling defined in that way would have lost that property too. The example also shows that also a labelling produced by $\mathcal{L}_{scc_{gr}}$ can maximise the set of in labelled argument. An example is also the floating assignment in Figure 2a, where argument c is accepted by the grounded SCC-void semantics but rejected by the preferred SCC-void semantics.

5 Related Works

The principles of beyond reasonable doubt, *in dubio pro reo* and standard of proof have been extensively study in argumentation theory (see [12]), but only few studies are relevant to abstract argumentation. In the context of structured argumentation, we mention the work by Prakken [17] on modelling standards of proof, and the modification of the Carneades framework [11].

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 skeptically preferred justification a beyond reasonable doubt position. In the floating assignment example (Figure 2a), the authors recognize the two attackers as doubtful, but they consider the skeptically 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 [12]). In case of beyond reason-able 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 [18], where he responds to objections advanced by Horty in [13]. 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 [16], the situation could be correctly modelled by making explicit some hidden assumptions 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 explored how the principles of *in dubio pro reo* and beyond reasonable doubt can be modelled in abstract argumentation. The two principles impose a high standard for an attack to be effective. We proposed to consider attacks from undecided arguments the weaker forms of attacks, considering them *not beyond reasonable doubt*. We proposed to neglect attacks from undecided arguments under some conditions and we generated the SCC-void semantics using an SCC-recursive schema. We provided a SCC-recursive version of SCC-void semantics and proved their fundamental proper-ties. 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 credulosly. We believe to have proposed an original and novel contribution to abstract argumentation semantics. Future research works include the investigation of alternative definitions of active and passive undecided argument, the study of the impact of the *in dubio pro reo* principle on the justification status of arguments and the study of how the principle could be implemented in probabilistic, fuzzy or numerical argumentation systems ([6][7][8]).

References

[1] Atkinson, K., and Bench-Capon, T. 2007. Argumentation and standards of proof. 11th international conference on Artificial intelligence and law, 107–116. ACM.
[2] Baroni, P.; Giacomin, M.; and Guida, G. 2005. Scc-recursiveness: a general schema for argumentation semantics. Artificial Intelligence 168(1-2):162–210.

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

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

[5] Caminada, M. W.; Carnielli, W. A.; and Dunne, P. E. 2012. Semi-stable semantics. Journal of Logic and Computation 22(5):1207–1254.

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

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

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

[9] Dondio, Pierpaolo. "Multi-Valued and Probabilistic Argumentation Frameworks." In COMMA, pp. 253-260. 2014.

[10] Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence 77(2):321–357.

[11] Gordon, T. F., and Walton, D. 2006. 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, volume 6, 5–13.

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

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

[14] Longo, L. and Hederman, L., 2013, October. Argumentation theory for decision support in health-care: a comparison with machine learning. In International Conference on Brain and Health Informatics (pp. 168-180). Springer, Cham.

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

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

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

[18] Prakken, H. 2002. Intuitions and the modelling of defeasible reasoning: some case studies. arXiv preprint cs/0207031.