On the Responsibility for Undecisiveness in Preferred and Stable Labellings in Abstract Argumentation (Extended Abstract)*

Claudia Schulz and **Francesca Toni**
Department of Computing, Imperial College London, UK
clauschulz1812@gmail.com, f.toni@imperial.ac.uk

Abstract

Different semantics of abstract Argumentation Frameworks (AFs) provide different levels of decisiveness for reasoning about the acceptability of conflicting arguments. The stable semantics is useful for applications requiring a high level of decisiveness, as it assigns to each argument the label "accepted" or the label "rejected". Unfortunately, stable labellings are not guaranteed to exist, thus raising the question as to which parts of AFs are responsible for the non-existence. In this paper, we address this question by investigating a more general question concerning preferred labellings (which may be less decisive than stable labellings but are always guaranteed to exist), namely why a given preferred labelling may not be stable and thus undecided on some arguments. In particular, (1) we give various characterisations of parts of an AF, based on the given preferred labelling, and (2) we show that these parts are indeed responsible for the undecisiveness if the preferred labelling is not stable. We then use these characterisations to explain the non-existence of stable labellings.

1 Introduction

Argumentation formalisms have been widely studied for representing arguments and conflicts between these arguments, and for evaluating which sets of arguments should be accepted by resolving the conflicts. An important application area of such formalisms is in decision support, where decisions are made based on an exchange of arguments and an evaluation of their acceptability (see for example [Kakas and Moraitis, 2003; Amgoud and Prade, 2009; Bench-Capon *et al.*, 2012; Fan *et al.*, 2014; Ferretti *et al.*, 2017]).

One of the most prominent formalisms is abstract *Argumentation Frameworks* (AFs) [Dung, 1995], assuming as given a set of arguments and attacks between them. AFs are equipped with different semantics, defining which arguments should be deemed acceptable. They can be defined in terms of acceptable sets of arguments (so called *extensions* [Dung, 1995]) or equivalently in terms of *labellings* [Caminada and Gabbay, 2009; Baroni *et al.*, 2011], which assign one of the labels `in` (accepted), `out` (rejected), or `undec` (undecided) to each argument. Different semantics impose different restrictions on labellings. Each argument needs to be *legally* labelled, where an `in`-labelled argument is legally labelled if all arguments attacking it are labelled `out`, an `out`-labelled argument is legally labelled if at least one argument attacking it is labelled `in`, and an `undec`-labelled argument is legally labelled if at least one argument attacking it is labelled `undec` and no argument attacking it is labelled `in` [Caminada and Gabbay, 2009; Baroni *et al.*, 2011].

In many applications, it is desirable to choose a highly decisive semantics, in other words, a semantics that assigns the label `in` or the label `out` to as many arguments as possible. Compared to less decisive semantics, this means greater certainty about the acceptance status of arguments for the user. In particular, the *preferred semantics* assigns the label `in` to a maximal set of arguments (w.r.t. set inclusion). If all arguments in a preferred labelling are labelled `in` or `out`, the labelling is *stable*. In applications requiring decisiveness, e.g. in medical or legal scenarios, it is desirable to have at least one stable labelling. Unfortunately, stable labellings are not guaranteed to exist, that is, in some cases all preferred labellings may comprise arguments labelled `undec`.

Running Example. As an illustration, consider the following example from the medical domain, represented graphically as an AF in Figure 1, where nodes are arguments and directed edges are attacks. A physician needs to decide which therapy amongst five possible therapies to recommend to her patient. She first reads a study praising therapy A and concluding that therapy A is way more effective than therapy B. This study thus provides an argument for the effectiveness of therapy A and positions it as a counterargument against any argument stating that therapy B is effective and should be chosen. In Figure 1, this is indicated by the attack from argument "A is effective" to argument "B is effective", which the physician obtains reading a second article. This second article recommends therapy B, showing that it is more reliable than therapy C and much more effective than therapy D. The physician reviews a third study, which describes the enormous success of therapy C and the poor performance of

*This extended abstract informally summarises the main contributions of the Artificial Intelligence article [Schulz and Toni, 2018].

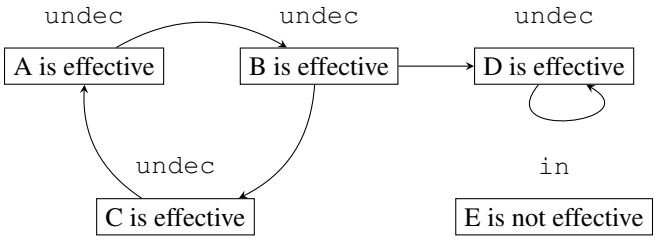

Figure 1: AF representing the physician's information about therapies and the AF's only preferred labelling *Pref*.

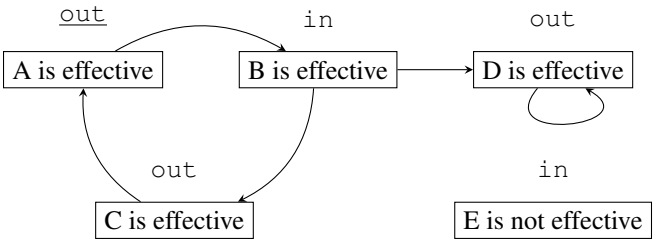

Figure 2: AF representing the physician's information, where undec labels from the preferred labelling are replaced by in or out labels (underlined labels are illegal).

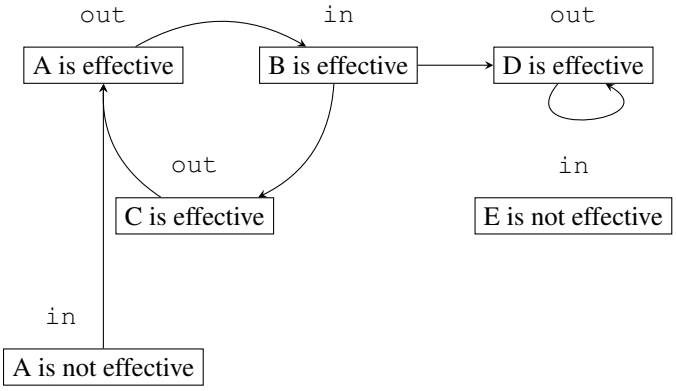

Figure 3: AF representing the physician's information, where the previously illegal out-label of the argument about therapy A is enforced by adding a new argument.

therapy A compared to C. Another article advocates therapy D somewhat incoherently, providing within the same study evidence against the effectiveness of this therapy. Therefore, the argument "D is effective" in Figure 1 attacks itself. Finally, a fifth article discusses therapy E, providing evidence against its effectiveness.

The resulting AF, representing the physician's information on the effectiveness of the five therapies, has a single preferred labelling but no stable labelling. Thus, using the stable semantics, no therapy can be recommended. The preferred labelling, referred to as *Pref* and given in Figure 1, labels all arguments as undec except for argument "E is not effective", which is labelled in. Thus, using the preferred semantics, the physician can draw the conclusion that therapy E is definitely not effective but still cannot make any decision as to which therapy to prescribe. Thus, the non-existence of stable labellings and the undecisiveness of preferred labellings are closely connected problems.

In the remainder, we use the therapy in an argument as the argument itself (so, for example, A will stand for the argument "A is effective").

2 Responsibility for Non-Stable Labellings

In this paper we present the main ideas of our work [Schulz and Toni, 2018] on the non-existence of stable labellings as a by-product of identifying, for a chosen non-stable preferred labelling of a given AF, which parts of the AF can be deemed responsible that this preferred labelling is not stable. Naively, the set of *all* undec arguments may be deemed responsible if a preferred labelling is not stable, since these are the arguments violating the definition of a stable labelling.

However, it may be possible to legally label some undec arguments as in or out in the AF. We can thus define *responsibility* based on only those arguments that cannot be legally labelled in or out. Informally, a set of arguments is deemed *responsible* if the AF requires some (structural) changes in order to turn the undec labels of arguments in this set into legal in or out labels.

In our running example, nearly all undec labels can be turned into legal in or out labels without any structural change, as illustrated in Figure 2. Only argument A requires a structural revision of the AF in order to turn its changed label out legal. Argument A can thus be deemed responsible that the preferred labelling is not stable. This structural change could be achieved, e.g., by adding a new argument attacking A, as illustrated in Figure 3. The new argument may be additional evidence found by the physician, concluding that therapy A is not effective at all.

In this paper, we are interested in the existence of some structural revision rather than its exact nature: the engineering of the revision is left open to fulfil differing requirements of applications and information available to users. We thus focus on the change of label from undec to in or out and the fact that *enforcing* the new label through some structural revision makes this label legal (in the structurally revised AF).

We propose two different characterisation approaches for identifying sets of responsible arguments: a *labelling-based* approach and a *structural* approach.

3 Labelling-Based Characterisations

In the labelling-based approach, we give characterisations of responsible parts using in-out *labellings* with respect to a preferred labelling *Pref*, that is, labellings that re-label all undec arguments in *Pref* as in or out and keep all in and out labels from *Pref* the same.

3.1 Enforcement Sets

Our first labelling-based characterisation defines *enforcement sets*. These are *minimal* sets of arguments labelled undec by *Pref* satisfying that some in-out labelling legally labels all non-responsible arguments (i.e. all arguments not contained in these enforcement sets).

In our running example, $\{A\}$ is an enforcement set w.r.t. *Pref*, as the labelling shown in Figure 2 is an `in-out` labelling that legally labels all arguments labelled `undec` by *Pref* except for argument A (i.e. arguments B, C, and D). Furthermore, $\{A\}$ is a minimal set satisfying this condition, since for its only subset $\{\}$ there exists no `in-out` labelling that legally labels *all* arguments labelled `undec` by *Pref*. There are two more enforcement sets w.r.t. *Pref*, namely $\{B\}$ and $\{C\}$. Note that $\{D\}$ is not an enforcement set since there exists no `in-out` labelling that legally labels A, B, and C. These enforcement sets are all disjoint. In general however, enforcement sets may contain some of the same arguments.

At least one enforcement set exists and enforcement sets are always non-empty if *Pref* is not stable. Both are important properties for sets of arguments characterising parts of an AF responsible that a preferred labelling is not stable.

The reason for naming this labelling-based characterisation "enforcement sets" is that "enforcing" the labels of an `in-out` labelling for arguments in an enforcement set in terms of a revision gives a stable labelling (of the revised AF), as illustrated in Figure 3. An enforcement set is thus *sufficient* for obtaining a stable labelling through a revision. It follows that an enforcement set consists of all arguments jointly responsible that a preferred labelling is not stable.

3.2 Preventing Sets

Enforcement sets characterise responsible sets with respect to a *specific* `in-out` labelling, which illegally labels *all* arguments in this set. An alternative labelling-based characterisation defines a responsible set of arguments as a *preventing set*, i.e. a minimal set containing at least *one* illegally labelled argument with respect to *every* `in-out` labelling.

The only preventing set w.r.t. *Pref* in our running example is $\{A, B, C\}$, since no matter how the labels `in` and `out` are assigned to this set of arguments, at least one argument is illegally labelled. In contrast, for all subsets there exists some `in-out` labelling that legally labels all arguments. For instance, for the set $\{A, B\}$, an `in-out` labelling that labels A as `in` and B and C as `out` legally labels both A and B.

As for enforcement sets, at least one preventing set exists w.r.t. any *Pref* and preventing sets are always non-empty if *Pref* is not a stable labelling.

The reason for naming this labelling-based characterisation "preventing sets" is that all (structural) revisions w.r.t. a set of arguments not comprising any argument from some preventing set have no stable labelling. In our running example, revising D will not result in a stable labelling. Thus, preventing sets define a *sufficient* condition for "preventing" the existence of a stable labelling.

Note that our labelling-based characterisations are defined with respect to any preferred labelling. For preferred labellings that are stable, the empty set of arguments is the only "responsible" set identified by either characterisation. We can therefore show that an AF has no stable labelling if and only if, with respect to all preferred labellings, there exists a non-empty set of arguments identified as the responsible part.

3.3 Enforcement vs. Preventing Sets

Enforcement and preventing sets characterise two sides of the same coin. In fact, a preventing set is a minimal set containing at least one argument from each enforcement set. Conversely, an enforcement set is a minimal set containing at least one argument from each preventing set.

This duality is mirrored in the type of responsibility sets they characterise. Enforcement sets are responsible since they consist of exactly the arguments whose labels need to be *enforced* in order to obtain a stable labelling, whereas preventing sets are responsible because they consist of exactly those arguments that *prevent* the existence of a stable labelling if the label of no argument in the set is enforced. Depending on the application at hand, one or the other type of characterisation may thus be more appropriate.

4 Structural Characterisations

Determining responsible sets of arguments according to the declarative labelling-based characterisations involves guessing sets of arguments and checking if they satisfy the respective definition. Instead, we can also characterise sets of arguments as responsible that a preferred labelling is not stable based on the *structure* of the AF. We thereby aim at characterisations that allow for a *constructive* determination of responsible sets of arguments.

4.1 Responsible Odd-Length Cycles

Our first structural characterisation is inspired by the seminal work of Dung [1995], who proved that if an AF has no odd-length cycles, then a stable labelling[1] exists. Consequently, the non-existence of stable labellings implies the existence of an odd-length cycle.

We show that, furthermore, an odd-length cycle exists if some preferred labelling is not stable, even if the AF has a stable labelling. In particular, there exists an odd-length cycle of arguments labelled `undec` by this (non-stable) preferred labelling. Thus, we define such odd-length cycles of arguments labelled `undec` as responsible that the preferred labelling is not stable. The reason to exclude odd-length cycles of arguments labelled `in` or `out` is that such cycles do not violate the definition of a stable labelling. In our running example, the cycle $A - B - C$ is thus a responsible cycle, whereas the cycle D is not.

In contrast to our labelling-based characterisations, which always exist but coincide with the empty set in case *Pref* is a stable labelling, responsible cycles exist if and only if *Pref* is not stable. Thus, responsible cycles are well-defined characterisations of parts of an AF responsible that *Pref* is not a stable labelling. They furthermore provide a *sufficient* condition to obtain a stable labelling by enforcing (suitably chosen) labels for arguments in all responsible cycles.

4.2 Strongly Connected `undec` Parts

An alternative structural characterisation identifies responsible parts as initial strongly connected components (SCCs)

[1] This follows from the correspondence between extensions and labellings [Caminada and Gabbay, 2009].

[Baroni *et al.*, 2005] of the AF restricted to arguments labelled undec by the preferred labelling. We call such parts *strongly connected* undec *parts* (SCUPs).

The only SCUP of the AF in Figure 1 is the cycle of arguments about therapies A, B, and C, so the set of these three arguments is deemed responsible by our structural approach that *Pref* is not a stable labelling. Here the only SCUP coincides with the only responsible cycle. We prove that, in general, every SCUP comprises a responsible cycle.

The notion of SCUPs is inspired by the decomposability result of Baroni et al. [2014], who show that the complete labellings of an AF can be obtained by splitting the AF into *any* partition and then determining complete labellings of the different parts so that they are compatible. We can thus think of *Pref* as a combination of two compatible labellings: a labelling of the part of the AF whose arguments are labelled in or out by *Pref*, and a labelling of the part of the AF whose arguments are labelled undec by *Pref*. We call these two parts the in/out-part and the undec-part, respectively.

The fact that all arguments in the undec-part are labelled undec by *Pref* implies that this is the only labelling compatible with the in and out labels in the in/out-part (if there was another labelling, *Pref* would not be maximal). We extend this result by proving that labelling all arguments in the undec-part as undec is the *only complete labelling* of this part on its own (disregarding the in/out-part). In other words, the labels of arguments in the in/out-part are not responsible that all arguments in the undec-part are labelled undec. Rather, the structure of the undec-part itself is responsible that the arguments cannot be legally labelled in or out. Since the undec-part has only one complete labelling, which labels all arguments as undec, this is also its only preferred labelling. Thus, the question as to why *Pref* is not a stable labelling can be reduced to the question as to why the only preferred labelling of the undec-part is not stable.

Another decomposability result states that stable labellings can be computed along the SCCs of the AF [Baroni *et al.*, 2005]. That is, the stable labellings of initial SCCs are computed and, subsequently, the stable labellings of the following SCCs are iteratively determined, while taking the labels of arguments in their parent SCCs into account. It follows that if an AF has no stable labelling, some SCC in this iterative computation has no stable labelling (when taking the labels in parent SCCs into account). Thus, the "first" SCCs with no stable labelling in the iterative computation of a stable labelling, given the labels of *Pref*, can be deemed responsible.

Applying this idea of responsible SCCs to the question why the only preferred labelling of the undec-part is not a stable labelling, we obtain that the reasons are its "first" SCCs that have no stable labelling. These "first" SCCs are the initial SCCs of the undec-part since *no* SCC in the undec-part has a stable labelling. This observation results in the above characterisation of SCUPs: an initial SCC of the undec-part.

SCUPs give a *necessary* condition for turning a non-stable into a stable labelling (via structural revision). We also show that iteratively enforcing arguments in SCUPs gives a sufficient condition for turning a non-stable into a stable labelling (via structural revision). SCUPs can thus be deemed responsible that the given preferred labelling is not stable.

5 Discussion

Having defined multiple characterisations of sets of arguments that are responsible that a preferred labelling is not stable, we now further investigate their relationship and show how they can explain the non-existence of stable labellings.

5.1 Labelling-Based versus Structural Characterisations

Despite the difference in the definitions of labelling-based and structural characterisations, the examples and the fact that all of them characterise responsible sets of arguments hint at similarities between them.

SCUPs and preventing sets share the property that if none of their arguments is revised, then no stable labelling is obtained, hinting at a close connection between the two characterisations. Indeed, we prove that each SCUP comprises a preventing set. In turn, each preventing set comprises a responsible cycle.

In general, SCUPs are not subsets of enforcement sets or vice versa. However, each SCUP contains an argument from each enforcement set. Furthermore, there exists an enforcement set that consists only of arguments from responsible cycles. Note that not every responsible cycle shares arguments with some enforcement set.

5.2 Non-Existence of Stable Labellings

Throughout this paper, we gave different characterisations of parts of an AF responsible that a given preferred labelling is not stable, irrespective of the existence of a stable labelling. That is, in general, the AF may have various preferred labellings, some that are stable and some that are not. These preferred labellings differ in their assignment of the labels in and out to certain arguments, in other words, an argument may be labelled in by one but out by another preferred labelling. This gives users the freedom to choose an assignment according to their own preferences.

In applications where decisiveness is required, users can thus decide whether they only care about finding *some* labelling without undec labels, in which case they can simply choose a stable labelling (if one exists), or they can choose one of the preferred labellings according to their preference concerning the assignment of in and out labels, and, if this preferred labelling is not stable, identify a suitable revision of the AF. If the AF has no stable labelling at all, the second situation is the only possible one. Our characterisations are thus versatile, as they can be applied in scenarios where an AF has no stable labelling and in scenarios where stable labellings exist, but the desired preferred labelling is not stable.

Since every stable labelling is a preferred labelling [Caminada and Gabbay, 2009], it follows that if no stable labelling exists, then no preferred labelling is stable. Thus, in the case of non-existence of stable labellings, our characterisations can explain the non-existence in terms of the preferred labellings not being stable.

Which of our characterisations is most suitable for an application in question is left to the user to decide. As we have shown, each characterisation defines parts of an AF that are indeed responsible that a preferred labelling is not stable, and consequently, if applicable, that no stable labelling exists.

References

[Amgoud and Prade, 2009] Leila Amgoud and Henri Prade. Using Arguments for Making and Explaining Decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.

[Baroni *et al.*, 2005] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-Recursiveness: A General Schema for Argumentation Semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005.

[Baroni *et al.*, 2011] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, 26(04):365–410, 2011.

[Baroni *et al.*, 2014] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert W. N. van der Torre, and Serena Villata. On the Input/Output Behavior of Argumentation Frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[Bench-Capon *et al.*, 2012] Trevor J. M. Bench-Capon, Katie Atkinson, and Peter McBurney. Using Argumentation to Model Agent Decision Making in Economic Experiments. *Autonomous Agents and Multi-Agent Systems*, 25(1):183–208, 2012.

[Caminada and Gabbay, 2009] Martin Caminada and Dov M. Gabbay. A Logical Account of Formal Argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[Dung, 1995] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–357, 1995.

[Fan *et al.*, 2014] Xiuyi Fan, Francesca Toni, Andrei Mocanu, and Matthew Williams. Dialogical Two-Agent Decision Making with Assumption-Based Argumentation. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14)*, pages 533–540, 2014.

[Ferretti *et al.*, 2017] Edgardo Ferretti, Luciano H. Tamargo, Alejandro J. García, Marcelo Luis Errecalde, and Guillermo R. Simari. An Approach to Decision Making based on Dynamic Argumentation Systems. *Artificial Intelligence*, 242:107–131, 2017.

[Kakas and Moraitis, 2003] Antonis C. Kakas and Pavlos Moraitis. Argumentation Based Decision Making for Autonomous Agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*, pages 883–890, 2003.

[Schulz and Toni, 2018] Claudia Schulz and Francesca Toni. On the responsibility for undecisiveness in preferred and stable labellings in abstract argumentation. *Artificial Intelligence*, 262:301 – 335, 2018.