An Argument Game for Stable Semantics

Martin Caminada[a], Yining Wu[b]

[a] *University of Luxembourg, martin.caminada@uni.lu*
[b] *University of Luxembourg, yining.wu@uni.lu*

Abstract

In this paper, we present a discussion game for argumentation under stable semantics. Our work is inspired by Vreeswijk and Prakken, who have defined a similar game for preferred semantics. In the current paper, we restate Vreeswijk and Prakken's work using the approach of argument labellings and then show how it can be adjusted for stable semantics. The nature of the resulting argument game is somewhat unusual, since stable semantics does not satisfy the property of *relevance*.

Key words: argumentation, argument labellings, discussion games, stable semantics

1. Introduction

Stable semantics, a concept that goes back to [18], is one of the oldest semantics for argumentation and non-monotonic reasoning. Although Dung's landmark paper [10] was partially meant to argue against the use of it, stable semantics has remained an important concept in fields like default logic [16] and logic programming [12, 13].

During recent years, several new semantics have been proposed [1, 6, 11, 2]. What makes stable semantics unique, however, are two fundamental properties. First of all, there is the possible absence of stable extensions. When applying stable semantics in, for instance, answer set programming, this can in fact be a desirable property. If one encodes a problem such that the possible solutions correspond with the stable extensions, then the absence of stable extensions indicates the absence of solutions to the original problem. Secondly, stable semantics does not satisfy the property of *relevance* [6]. That is, it is possible for the status of an argument A to be influenced by a totally unrelated argument B. For instance, let (Ar, def) be an argumentation framework where the set of arguments Ar is $\{A, B\}$ and the defeat relation def is $\{(B, B)\}$, meaning that B defeats B. Then A and B are totally unrelated in the sense that there does not exist an (undirected) defeat-path between A and B. Yet, the existence of argument B causes argument A not to be credulously accepted.

The invalidity of the property of relevance has implications for the possibilities of defining an argument game.[1] For instance, for grounded and preferred semantics, both of which do satisfy relevance, it is possible to define argument games in which each move is a response to a previous move [19, 15, 4]. For stable semantics, however, this is not possible. In the above example, argument B is the reason why argument A is not credulously accepted. Yet, it would be somewhat odd to reply to A with B, since no relation exists between these arguments.

In this paper, we propose an argument game that can deal with the unique characteristics of stable semantics. First, in Section 2, we briefly state some preliminaries on argument semantics and argument labellings. Then, in Section 3, we restate the approach of Vreeswijk and Prakken in terms of argument labellings. The discussion game for credulous acceptance under stable semantics is then given in Section 4, and an approach for sceptical acceptance under stable semantics is given in Section 5. Then, in Section 6, we finish with a discussion about some future research topics.

2. Argument Semantics and Argument Labellings

In this section, we briefly restate some preliminaries regarding argument semantics and argument-labellings. We restrict ourselves to finite argumentation frameworks in order to assure termination for the proposed discussion game defined later in this paper.

Definition 1. *An* argumentation framework *is a pair* (Ar, def) *where* Ar *is a finite set of arguments and* $def \subseteq Ar \times Ar$*.*

We say that argument A *defeats* argument B (or alternatively, that A is a *defeater* if B) iff $(A, B) \in def$.

An argumentation framework can be represented as a directed graph in which the arguments are represented as nodes and the defeat relation is represented as arrows. In several examples throughout this paper, we will use this graph representation.

Definition 2 (defense / conflict-free)**.**
Let (Ar, def) *be an argumentation framework,* $A \in Ar$ *and* $\mathcal{A}rgs \subseteq Ar$*.*
We define A^+ *as* $\{B \mid A \ def \ B\}$ *and* $\mathcal{A}rgs^+$ *as* $\{B \mid A \ def \ B$ *for some* $A \in \mathcal{A}rgs\}$*.*
We define A^- *as* $\{B \mid B \ def \ A\}$ *and* $\mathcal{A}rgs^-$ *as* $\{B \mid B \ def \ A$ *for some* $A \in \mathcal{A}rgs\}$*.*
$\mathcal{A}rgs$ *is* conflict-free *iff* $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$*.*
$\mathcal{A}rgs$ defends *an argument* A *iff* $A^- \subseteq \mathcal{A}rgs^+$*.*
Let $F : 2^{Ar} \rightarrow 2^{Ar}$ *be the function defined as:*
$F(\mathcal{A}rgs) = \{A \mid A$ *is defended by* $\mathcal{A}rgs\}$*.*

[1] An argument game (in the sense of [19, 15, 4]) can be described as a formal discussion in which two parties (proponent and opponent) take turns to exchange arguments. Usually, each of their moves consists of uttering an argument that defeats one of the other party's previous arguments. The ultimate aim of the game is to determine whether the main argument (uttered in the first move by the proponent) can be considered to be justified.

In the definition below, stable semantics is described as an admissible set that defeats each argument that is not an element of it. It can be proven that this is the same as Dungs original definition [10] of a stable extension as a conflict-free set that defeats each argument that is not an element of it

Definition 3 (acceptability semantics)**.** *Let* (Ar, def) *be an argumentation framework. A conflict-free set* $\mathcal{A}rgs \subseteq Ar$ *is called*

- *an* admissible set *iff* $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$.
- *a* preferred extension *iff* $\mathcal{A}rgs$ *is a maximal admissible set.*
- *a* stable extension *iff* $\mathcal{A}rgs$ *is an admissible set that defeats every argument in* $Ar \backslash \mathcal{A}rgs$.

The concepts of admissibility, as well as those of preferred and stable semantics were originally stated in terms of sets of arguments. It is equally well possible, however, to express these concepts using *argument labellings*. This approach has been proposed by Pollock [14] and Verheij [17] and has recently been extended by Caminada [5]. The idea of a labelling is to associate with each argument at most one label, which can be `in` or `out`. It is also possible for an argument to be unlabelled.[2] The label `in` indicates that the argument is explicitly accepted and the label `out` indicates that the argument is explicitly rejected.

Definition 4. *A* labelling *is a partial function* $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}\}$.

We write $\texttt{in}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{in}\}$ and $\texttt{out}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{out}\}$.

Since a labelling is a function, which is essentially a relation, it can be represented as a set of pairs. For instance, a possible labelling of the argumentation framework of Figure 1 would be $\{(A, \texttt{in}), (B, \texttt{out})\}$

Definition 5. *Let* $\mathcal{L}$ *be a labelling of argumentation framework* (Ar, def) *and* $A \in Ar$. *We say that:*

1. *A is* legally `in` *iff A is labelled* `in` *and each defeater of A is labelled* `out`
2. *A is* legally `out` *iff A is labelled* `out` *and A has at least one defeater that is labelled* `in`

We say that an argument A is *illegally* `in` iff A is labelled `in` but is not legally `in`. We say that an argument A is *illegally* `out` iff A is labelled `out` but is not legally `out`.

Definition 6. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}\}$ *be a partial function. We say that* $\mathcal{L}$ *is an* admissible labelling *iff it satisfies the following:*

[2] In [5, 7] an argument has exactly one label, which is either `in`, `out` or `undec`. This approach is suitable for capturing the notion of complete semantics. However, in the current paper, we are merely interested in capturing the notion of preferred and stable semantics, for which the current, simple definition of a labelling is sufficient.

- *each argument that is labelled* in *is legally* in*.*
- *each argument that is labelled* out *is legally* out*.*

A preferred labelling *is a maximal (w.r.t. set inclusion) admissible labelling $\mathcal{L}$. A* stable labelling *is an admissible labelling $\mathcal{L}$ where each argument is labeled.*

As an example, consider the argumentation framework of Figure 1. Here $\emptyset, \{(A, \texttt{in}), (B, \texttt{out})\}, \{(A, \texttt{out}), (B, \texttt{in})\}$ and $\{(A, \texttt{out}), (B, \texttt{in}), (C, \texttt{out}), (D, \texttt{in}), (E, \texttt{out})\}$ are examples of admissible labellings. Only $\{(A, \texttt{in}), (B, \texttt{out})\}$ and $\{(A, \texttt{out}), (B, \texttt{in}), (C, \texttt{out}), (D, \texttt{in}), (E, \texttt{out})\}$ are preferred labellings (because they are maximal w.r.t. set inclusion). Only $\{(A, \texttt{out}), (B, \texttt{in}), (C, \texttt{out}), (D, \texttt{in}), (E, \texttt{out})\}$ is a stable labelling.

It can be proved that the various types of labellings correspond to the various kinds of argument semantics [5, 7]. The first step is to show that admissible sets coincide with admissible labellings. This can then serve as the basis to show that preferred extensions coincide with preferred labellings and that stable extensions coincide with stable labellings.

Theorem 1. *Let (Ar, def) be an argumentation framework and let $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is an admissible set iff there exists an admissible labelling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$.*

Proof.
"$\Longrightarrow$": Let Ar be an admissible set. Now consider a labelling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$ and $\texttt{out}(\mathcal{L}) = \mathcal{A}rgs^+$. From the fact that $\mathcal{A}rgs$ is conflict-free it follows that $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$, so the labelling is well-defined.
We now prove that $\mathcal{L}$ has no arguments that are illegally in. Let A be an arbitrary argument that is labelled in by $\mathcal{L}$. Then $A \in \mathcal{A}rgs$. Let B be an arbitrary defeater of A. The fact that $\mathcal{A}rgs$ is an admissible set means that $\mathcal{A}rgs$ contains an argument (say C) that defeats B. Therefore, $B \in \mathcal{A}rgs^+$, so B is labelled out by $\mathcal{L}$. As this holds for any arbitrary defeater B of A, it follows that each defeater of A is labelled out. Therefore, A is legally in.
Next we prove that $\mathcal{L}$ has no arguments that are illegally out. Let A be an arbitrary argument that is labelled out by $\mathcal{L}$. Then $A \in \mathcal{A}rgs^+$. The fact that $A \in \mathcal{A}rgs^+$ means that there is some argument (say B) in $\mathcal{A}rgs$ that defeats A. The fact that B is in $\mathcal{A}rgs$ means that B is labelled in by $\mathcal{L}$. Thus, A has a defeater that is labelled in. Therefore, A is legally out.
"$\Longleftarrow$": Let $\mathcal{L}$ be an admissible labelling and let $\mathcal{A}rgs = \texttt{in}(\mathcal{L})$.
We first prove that $\mathcal{A}rgs$ is conflict-free. Suppose this is not the case. Then there exist two (possibly the same) arguments $A, B \in \mathcal{A}rgs$ such that A defeats B. The fact that $A, B \in \mathcal{A}rgs$ means that both A and B are labelled in by $\mathcal{L}$. But then B would be illegally in, which implies that $\mathcal{L}$ is not an admissible labelling. Contradiction.
We now prove that $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$, Let $A \in \mathcal{A}rgs$. Then A is labelled in by $\mathcal{L}$. Let B be an arbitrary defeater of A. The fact that $\mathcal{L}$ is an admissible labelling means that $\mathcal{L}$ has no arguments that are illegally in, so from the fact that A is labelled in by $\mathcal{L}$ it follows that B must be labelled out. The fact that $\mathcal{L}$ is an

admissible labelling also means that $\mathcal{L}$ has no arguments that are illegally `out`, so from the fact that B is labelled `out` it follows that B must have a defeater (say C) that is labelled `in` means that $C \in \mathcal{A}rgs$. So, as A is defended by $\mathcal{A}rgs$ against any possible defeater B, we have that $A \in F(\mathcal{A}rgs)$. □

Theorem 2. *Let (Ar, def) be an argumentation framework and let $\mathcal{A}rgs \subseteq Ar$.*
$\mathcal{A}rgs$ is a preferred extension iff
there exists a preferred labelling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$.
$\mathcal{A}rgs$ is a stable extension iff
there exists a stable labelling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$.

Proof. Using the results of Theorem 1 this then follows in a straightforward way from Definition 2 and Definition 6. □

There are different ways to characterize a stable extension.

Proposition 1. *Let (Ar, def) be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. The following statements, describing the concept of stable semantics, are equivalent:*

1. *$\mathcal{A}rgs$ defeats exactly the arguments in $Ar \backslash \mathcal{A}rgs$*
2. *$\mathcal{A}rgs$ is a conflict-free set*
   that defeats each argument in $Ar \backslash \mathcal{A}rgs$
3. *$\mathcal{A}rgs$ is an admissible set*
   that defeats each argument in $Ar \backslash \mathcal{A}rgs$
4. *$\mathcal{A}rgs$ is a preferred extension*
   that defeats each argument in $Ar \backslash \mathcal{A}rgs$

3. Vreeswijk and Prakken's Argumentation Game for Preferred Semantics

In this section we treat a reformulated version of Vreeswijk and Prakken's argument game for preferred semantics [19]. Although there also exist other argument games for preferred semantics, like [8], we have chosen [19] for its relative simplicity and its easy adaptability to work with argument labellings. Our reformulation is aimed at slightly simplifying Vreeswijk and Prakken's approach, and also to allow for its easy adaptation to stable semantics, which will be treated in the next section.

In order to determine whether an argument (say A) is in an admissible set (say $\mathcal{A}rgs$), one can examine whether there exists an admissible labelling ($\mathcal{L}$) with $\mathcal{L}(A) = \texttt{in}$ (Theorem 2). The discussion game is then aimed at providing this admissible labelling. The admissible discussion game can be described as follows:

- proponent (P) and opponent (O) take turns; P begins
- each move of O is a defeater of some (not necessarily the directly preceding) previous argument of P

- each move of P (except the first one) is a defeater of the directly preceding argument of O
- O is not allowed to repeat its own moves, but may repeat P's moves
- P is not allowed to repeat O's moves, but may repeat its own moves

The game is won by the proponent iff the opponent cannot move anymore. It is won by the opponent iff the proponent cannot move anymore, or if the opponent manages to repeat one of the proponent's moves.

One good way to view the discussion game is as the proponent trying to build the set of `in`-labelled arguments and the opponent trying to build the set of `out`-labelled arguments. As an example, consider the argumentation framework illustrated in Figure 1.

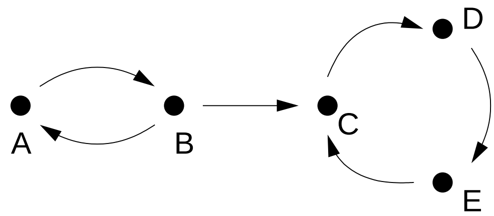

Figure 1: An argumentation framework

Here, the proponent can win the discussion game for argument D in the following way:
P: `in`(D) "I have an admissible labelling in which D is labelled `in`."
O: `out`(C) "Then in your labelling it must also be the case that D's defeater C is labelled `out` (otherwise D would not be legally `in`). Based on which grounds?"
P: `in`(B) "C is labelled `out` because B is labelled `in`."
O: `out`(A) "Then in your labelling it must also be the case that B's defeater A is labelled `out` (otherwise B would not be legally `in`). Based on which grounds?"
P: `in`(B) "A is labelled `out` because B is labelled `in`."

The above example illustrates the need for the proponent to be able to repeat its own arguments. At the same time, the proponent should not be allowed to repeat the opponent's arguments, since these have to be labelled `out`, so the proponent cannot claim them to be labelled `in`.

The argumentation framework of Figure 1 can also be used for an example of a game won by the opponent:
P: `in`(E) "I have an admissible labelling in which E is labelled `in`."
O: `out`(D) "Then in your labelling it must be the case that E's defeater D is labelled `out`. Based on which grounds?"
P: `in`(C) "D is labelled `out` because C is labelled `in`."
O: `out`(E) "Then in your labelling it must be the case that C's defeater E is

labelled `out`. This contradicts with your earlier claim that E is labelled `in`."

The above example illustrates that it is necessary to allow the opponent to repeat the proponent's arguments. Nevertheless, it would not be useful for the opponent to repeat its own arguments, since the reason why the particular argument is labelled out has already been explained by the proponent, so it makes no sense to ask again.

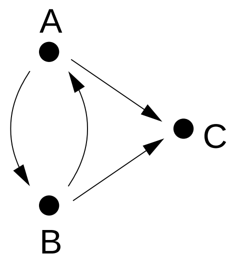

Figure 2: An argumentation framework with floating defeat

As a last illustration of the dialogue game for admissibility, consider the argumentation framework of Figure 2. Argument C is not in an admissible set. It is illustrative to see what happens if the proponent tries to defend C.
P: `in`(C) "I have an admissible labelling in which C is labelled `in`."
O: `out`(A) "Then in your labelling C's defeater A must be labelled `out`. Based on which grounds?"
P: `in`(B) "A is labelled `out` because B is labelled `in`."
O: `out`(B) "But from the fact that you hold C to be `in`, it follows that C's defeater B must be labelled `out`. This contradicts with your earlier claim that B is labelled `in`."

The above example illustrates the need for the opponent to be able to respond not only to the immediately preceding move, but to any past move of the proponent; in the example, `out`(B) is a response to `in`(C). This is because for an argument to be legally `in`, *all* its defeaters have to be `out`, so the opponent may need to respond to the proponent's argument with more than one move. At the other hand, it is not needed for the proponent to be able to respond to any move but the previous one. This is because for an argument to be legally `out` it is sufficient to have one defeater that is labelled `in`. If the proponent simply gives this defeater, then there is no need to give any additional defeaters. Hence, the proponent only needs to respond to the directly preceding move of the opponent.

The correlation between the thus described discussion game and the concept of admissibility can be described as follows.

Theorem 3. *Let* (Ar, def) *be an argumentation framework and* $A \in Ar$. *There exists an admissible labelling* $\mathcal{L}$ *with* $\mathcal{L}(A) =$ `in` *iff there exists an admissible* *discussion for A that is won by the proponent.*

Proof.
"$\Longrightarrow$": Suppose there exists an admissible labelling that labels A `in`. Now, consider a discussion that is started with A. It is possible for the proponent only to give moves that are labelled `in` in $\mathcal{L}$. If the proponent adopts this strategy, then the opponent can only put forward moves that are labelled `out` in $\mathcal{L}$. Thus, the opponent is not able to repeat any of the proponent's arguments. Furthermore, the discussion will terminate when the opponent cannot move anymore, because it has played all possible defeaters of the arguments labelled `in` in $\mathcal{L}$.
"$\Longleftarrow$": Suppose there exists a discussion for A that is won by the proponent. Then the labelling $\mathcal{L}$ that labels all proponent-moves `in` and all opponent-moves `out` is an admissible labelling. This can be seen as follows. First of all, $\mathcal{L}$ is well-defined, since the fact that it is won by the proponent means there is no argument that is put forward by both the proponent and opponent. Also, for each argument that is labelled `in`, it holds that all its defeaters are labelled `out`. This follows from the fact that the opponent cannot move anymore, so every defeater of the proponent's arguments has been put forward by the opponent. Furthermore, each argument that is labelled `out` has at least one defeater that is labelled `in`. This is because each move of the opponent has been responded to by the proponent. □

Since the concept of admissible labellings coincides with the concept of an admissible set (theorem 2) it holds that an argument is in an admissible set iff it is possible for the proponent to win the discussion for it. Moreover, it holds that an argument is in an admissible set iff it is in a preferred extension (or, alternatively, iff it is labelled `in` in a preferred labelling). Hence, the discussion game can be used as a basis for proof procedures for credulous preferred.

Vreeswijk and Prakken show that the discussion game can also be used as a basis for the decision problem of sceptical preferred semantics. This approach, however, only works for argumentation frameworks where every preferred extension is also a stable extension.

4. A Discussion Game for Credulous Stable Semantics

In the current section, we provide the main result of this paper, which is a discussion game for credulous stable semantics. Before doing so, it may be illustrative to see why the standard admissibility discussion game does not work for stable semantics. Consider again the argumentation framework of Figure 1. Even though A is in an admissible set and in a preferred extension ($\{A\}$), A is not in a stable extension. To see why A is in an admissible set, consider the following discusion:
P: `in`(A) "I have an admissible labelling where A is labelled `in`"
O: `out`(B) "Then in your labelling, argument B must be labelled `out`. Based on which grounds?"

P: `in`(A) "B is labelled `out` because A is labelled `in`"
The point is, however, that once it has been committed that A is labelled `in` and B is labelled `out`, it is not possible anymore to label the remaining arguments such that final result will be a stable labelling. This can be seen as follows. Suppose C is labelled `in`. Then E must be labelled `out`, so D should be labelled `in`, which means that C would be labelled `out`. Contradiction. Similarly, suppose that C is labelled `out`. Then E must be labelled `in`, so D should be labelled `out`, so C should be labelled `in`. Again, contradiction.

Proposition 1 shows that there are many ways to characterize a stable extension. For our purposes, the most useful characterization is that of an admissible set which defeats every argument that is not in it. When one translates this to labellings, one obtains an admissible labelling where each argument is labelled either `in` or `out`, and no argument is left unlabelled.

It appears that a discussion game for stable semantics requires an additional type of move: `question`. To illustrate the role of this new move, imagine a politician being interviewed for TV. At first the discussion may be about financial matters (say, whether the banking system should be nationalized). Then, the discussion may be about the consequences of the politician's opinion ("If you accept to nationalize the banks, then you must reject the possibility to improve healthcare, because there will not be enough money left to do so."). However, at some moment, the interviewer could choose to totally change topic ("By the way, what are your opinions about abortion?"). It is this change of topic that is enabled by the `question` move.

For the discussion game for stable semantics, we use the `question` move to involve those arguments that have never been uttered before so that we can label all the elements of Ar. By questioning an argument (`question`(A)), the opponent asks the proponent to give an explicit opinion on whether A should be labelled `in` or `out`. If the proponent thinks that A should be labelled `in` then it should respond with `in`(A). If the proponent thinks that A should be labelled `out` then it should respond with `in`(B) where B is a defeater of A. The discussion game for stable semantics can thus be described as follows:

- The proponent (P) and opponent (O) take turns. The proponent begins.
- Each move of the opponent is either of the form `out`(A), where A is a defeater of some (not necessarily the directly preceding) move of the proponent, or of the form `question`(A), where A is an argument that has not been uttered in the discussion before (by either the proponent or the opponent). The opponent is only allowed to do a `question` move if it cannot do an `out` move.
- The first move of the proponent is of the form `in`(A), where A is the main argument of the discussion. The following moves of the proponent are also of the form `in`(A) although A no longer needs to be the main claim. If the directly preceding move of the opponent is of the form `out`(B) then A is a defeater of B. If the directly preceding move of the opponent is of the form `question`(B) then A is either equal to B or a defeater of B.

- The opponent may not repeat any of its `out` moves.
- The proponent is allowed to repeat its own moves, but may not do an `in`(A) move if the opponent has done some earlier `out`(A) move.

The opponent wins if it is able to do an `out`(A) move and the proponent has done an earlier `in`(A) move, or if the proponent cannot move anymore. The proponent wins if the opponent cannot move anymore.

To illustrate the use of the discussion game, consider the argumentation framework depicted in Figure 3.

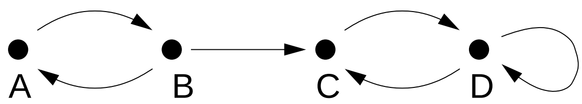

Figure 3: Another argumentation framework

Suppose the proponent would like to start a discussion about A.
P: `in`(A) "I have a stable labelling in which A is labelled `in`."
O: `out`(B) "Then in your labelling, A's defeater B must be labelled `out`. Based on which grounds?"
P: `in`(A) "B is labelled `out` because A is labelled `in`."
O: `question`(C) "What about C?"
P: `in`(C) "C is labelled `in`."
O: `out`(D) "Then C's defeater D must be labelled `out`. Based on which grounds?"
P: `in`(C) "D is labelled `out` because C is labelled `in`."
The proponent wins the discussion, since the opponent cannot move anymore.

The above example also shows that the outcome of a discussion may depend on P's response to a question move. For instance, if P would have replied to `question`(C) with `in`(D), then it would have lost the discussion, since O would then do `out`(D).

As an example of a game that cannot be won by the proponent, consider a game for argument B. This game has to be lost by the proponent since the argumentation framework of Figure 3 has only one stable extension: $\{A, C\}$, which does not include B.
P: `in`(B) "I have a stable labelling in which B is labelled `in`."
O: `out`(A) "Then in your labelling, B's defeater A must be labelled `out`. Based on which grounds?"
P: `in`(B) "A is labelled `out` because B is labelled `in`."
O: `question`(C) "What about C?"
P: `in`(D) "C is labelled `out` because its defeater D is labelled `in`."
O: `out`(D) "Then D's defeater D (itself) must be labelled `out`. Contradiction."
The proponent would still not have won the discussion if it had responded to `question`(C) with `in`(C) instead of with `in`(D). This is because then the opponent would have reacted with `out`(B) and would therefore still have won the discussion.

Formally, the stable discussion game can be described as follows.

Definition 7. *Let (Ar, def) be an argumentation framework. A* stable discussion *is a sequence of moves $[M_1, M_2, \ldots, M_n]$ $(n \geq 0)$ such that:*

- *each M_i $(1 \leq i \leq n)$ where i is odd (which is called a* proponent move*) is of the form* $\texttt{in}(A)$*, where $A \in Ar$.*
- *each M_i $(1 \leq i \leq n)$ where i is even (which is called an* opponent move*) is of the form* $\texttt{out}(A)$ *where $A \in Ar$, or of the form* $\texttt{question}(A)$ *where $A \in Ar$.*
- *For each opponent move $M_i = \texttt{out}(A)$ $(2 \leq i \leq n)$ there exists a proponent move $M_j = \texttt{in}(B)$ $(j < i)$ where A defeats B.*
- *For each proponent move $M_i = \texttt{in}(A)$ $(3 \leq i \leq n)$ it either holds that (1) $M_{i-1} = \texttt{out}(B)$ where A defeats B, or (2) $M_{i-1} = \texttt{question}(B)$ where either $A = B$ or A defeats B.*
- *For each opponent move $M_i = \texttt{out}(A)$ $(1 \leq i \leq n)$ there does not exist an opponent move $M_j = \texttt{out}(A)$ with $j < i$.*
- *For each opponent move $M_i = \texttt{question}(A)$ $(1 \leq i \leq n)$ there does not exist any move M_j $(j < i)$ of the form $\texttt{in}(A)$, $\texttt{out}(A)$ or $\texttt{question}(A)$.*
- *For each proponent move $M_i = \texttt{in}(A)$ $(1 \leq i \leq n)$ there does not exist an opponent move $M_j = \texttt{out}(A)$ with $j < i$.*

A stable discussion $[M_1, M_2, \ldots, M_n]$ is said to be finished *iff there exists no M_{n+1} such that $[M_1, M_2, \ldots, M_n, M_{n+1}]$ is a stable discussion, or if M_n is an opponent move of the form $\texttt{out}(A)$ for which there exists a proponent move M_i $(1 \leq i \leq n)$ of the form $\texttt{in}(A)$. A finished discussion is won by the proponent if the last move is a proponent move, and is won by the opponent if the last move is an opponent move.*

It turns out that an argument is in at least one stable extension iff the proponent can win the stable discussion game for it. Notice that the proof of this depends on the condition that the argumentation framework is finite, because the game requires each and every argument to have been played (either by the proponent or by the opponent) before it is finished.

Theorem 4. *Let (Ar, def) be an argumentation framework and $A \in Ar$. There exists a stable labelling $\mathcal{L}$ with $\mathcal{L}(A) = \texttt{in}$ iff there exists a stable discussion for A that is won by the proponent.*

Proof.
"$\Longrightarrow$": Suppose there exists a stable labelling $\mathcal{L}$ with $\mathcal{L}(A) = \texttt{in}$.
As the first step the discussion game the proponent utters $\texttt{in}(A)$. Trivially, it now holds that we have a discussion in which all $\texttt{in}$-labelled moves are also labelled $\texttt{in}$ in the stable labelling $\mathcal{L}$.

We now prove that any unfinished discussion where the proponent does the last move and where all proponent-moves are labelled `in` in $\mathcal{L}$ can be extended to a discussion with an additional opponent move and an additional proponent-move such that the result will again be a discussion in which the proponent does the last move, and all proponent moves are labelled `in` in $\mathcal{L}$.

Let $[M_1, \ldots, M_n]$ be an unfinished discussion where M_n is a proponent move and all proponent moves are labelled `in` in $\mathcal{L}$. From the fact that the discussion is unfinished, it follows that the opponent can do a move M_{n+1} which is either of the form $\texttt{out}(B)$, where B is a defeater of some earlier proponent move (say $\texttt{in}(A)$), or of the form $\texttt{question}(B)$. In the first case ($\texttt{out}(B)$), it holds that B is labelled `out` in $\mathcal{L}$, because A is labelled `in` in $\mathcal{L}$. It then follows that there exists an argument C which defeats B and is labelled `in` in $\mathcal{L}$, which makes it possible for the proponent to respond with $\texttt{in}(C)$. In the second case ($\texttt{question}(B)$), the proponent's response depends on whether B is labelled `in` or `out` in $\mathcal{L}$. If B is labelled `in` in $\mathcal{L}$, then the proponent replies with $\texttt{in}(B)$. However, if B is labelled `out` in $\mathcal{L}$, then from the fact that $\mathcal{L}$ is a stable (and therefore also admissible) labelling it follows that B has at least one defeater (say C) that is labelled `in` in $\mathcal{L}$. The proponent then replies with $\texttt{in}(C)$. In any case, the resulting discussion will have a proponent move as the last move, and all proponent-moves labelled `in` in $\mathcal{L}$.

From the fact that the argumentation framework is finite (Definition 1) and the fact that the opponent cannot repeat its moves, it follows that each discussion will ultimately finish. From the fact that every unfinished discussion game can always be extended to a discussion game in which the last move is still move by the proponent, it then follows that it is possible to play the game in such a way that when it is finished, the last move will be a proponent move, and that therefore the game will be won by the proponent.

"$\Longleftarrow$": Suppose there exists a stable discussion game for argument A that is won by the proponent. Let $\mathcal{A}rgs$ be the set of the `in` labelled arguments. $\mathcal{A}rgs$ is conflict-free, otherwise the opponent would have labelled an argument `out` that was labelled `in` by proponent earlier and would have won the game. Furthermore, $\mathcal{A}rgs$ defeats each argument that is not in $\mathcal{A}rgs$. This can be seen as follows. Let $B \notin \mathcal{A}rgs$. This implies that there has not been a proponent-move $\texttt{in}(B)$. From the fact that the discussion is finished it follows that the opponent has played each and every possible move. The fact that the proponent did not play $\texttt{in}(B)$ then implies that the opponent played $\texttt{out}(B)$ or $\texttt{question}(B)$. In the former case (the opponent played $\texttt{out}(B)$) it follows that the proponent reacted with a move $\texttt{in}(C)$ where C is a defeater of B. In the latter case (the opponent played $\texttt{question}(B)$) it follows that the proponent reacted with a move $\texttt{in}(C)$ where C is a defeater of B (the proponent did not react with $\texttt{in}(B)$ because then $B \in \mathcal{A}rgs$). So in either case (former or latter) the proponent reacted with $\texttt{in}(C)$ where C is a defeater of B. So $\mathcal{A}rgs$ contains an argument (C) that defeats B.

Since $\mathcal{A}rgs$ is conflict-free and defeats each argument not in it, $\mathcal{A}rgs$ is a stable extension. From Theorem 2, it follows that there exists a stable labelling with A is labelled `in`. □

For the discussion game for preferred semantics, it is quite straightforward to convert the resulting game to an admissible labelling: $\mathcal{L} = \{(A, \texttt{in}) \mid$ there exists a proponent-move $\texttt{in}(A)\} \cup \{(A, \texttt{out}) \mid$ there exists an opponent-move $\texttt{out}(A)\}$.

For the discussion game for stable semantics, converting the moves of the game to a stable labelling is slightly different. $\mathcal{L} = \{(A, \texttt{in}) \mid$ there exists a proponent-move $\texttt{in}(A)\} \cup \{(A, \texttt{out}) \mid$ there exists an opponent-move $\texttt{out}(A)\} \cup \{(A, \texttt{out}) \mid$ there exists an opponent-move $\texttt{question}(A)$ that was responded to with $\texttt{in}(B)$ where B is a defeater of $A\}$.

There are some possible optimizations for the above mentioned discussion game. As Vreeswijk and Prakken point out, the role of the opponent can also be seen as actually *helping* the proponent to find what it is looking for. If one takes this perspective, then it is quite reasonable to require the opponent to do a `question`-move only when it has (temporarily) cannot do an `out` move anymore. There is, however, another way in which the opponent can help the proponent to construct a stable labelling. If the opponent has to do a `question`-move, because it (temporarily) ran out of `out`-moves, then it makes sense for the opponent to try to do a $\texttt{question}(A)$-move such that (1) A is an argument that has a defeater that is `in` (that is, there exists an argument B such that B defeats A and the proponent did an $\texttt{in}(B)$-move in the past) or (2) A is an argument that has all its defeaters `out` (that is, for each argument B such that B defeats A, the opponent did either an $\texttt{out}(B)$-move or a $\texttt{question}(B)$-move at which the proponent did not respond with an $\texttt{in}(B)$-move). In both cases, it is clear how the proponent should respond. In case (1), the proponent should respond with $\texttt{in}(B)$ (where B is a defeater of A that was already found to be `in`; basically, the proponent is repeating one of its earlier moves). In case (2), the proponent will respond with $\texttt{in}(A)$. In general, the opponent could adapt a strategy of trying to select questions that are relatively easy to answer for the proponent. Such a strategy does not influence the correctness and completeness of the discussion game as a proof theory for stable semantics because it just gives the priority to some arguments to be questioned first.

5. A Discussion Game for Sceptical Stable Semantics

In [19] Vreeswijk and Prakken provide a procedure for determining if an argument is an element of every preferred extension. Their procedure, however, only works for argumentation frameworks where each preferred extension is also a stable extension. The discussion procedure for sceptical stable semantics that is proposed in this section does not have this restriction.

The idea is that an argument is in each stable extension iff there is no stable extension that contains one of its defeaters.

Theorem 5. *Let (Ar, def) be an argumentation framework, and let $A \in Ar$. A is an element of each stable extension iff there exists no stable extension containing a defeater of A.*

Proof.
"$\Longrightarrow$": Suppose A is an element of each stable extension. Then there exists no stable extension S that does not contain A. Therefore (since for each argument $A \in Ar$, a stable extension contains either A or a defeater of A), there exists no stable extension that contains a defeater of A.
"$\Longleftarrow$": Suppose there exists no stable extension containing a defeater of A. Then each stable extension does not contain a defeater of A. Therefore (since for each argument $A \in Ar$, a stable extension either contains A or a defeater of A) each stable extension contains A. □

So in order to examine whether an argument A is in each stable extension, one should examine the defeaters of A one by one. If one finds a defeater that is in a stable extension, then the question of whether A is in each stable extension can be answered with "no". If, however, it turns out that each defeater of A is not in any stable extension, then the answer is "yes". Therefore, one can simply apply the (credulous) stable discussion game for each defeater of A, to obtain the answer regarding sceptical stable.

6. Discussion and Further Research

In this paper, we have introduced a discussion game for stable semantics, based on the work of Vreeswijk and Prakken [19]. Our discussion game is not the only approach that can be based their work. The proof procedures of Dung, Mancarella and Toni for *ideal semantics* [11] can, for instance, also be described in terms of Vreeswijk and Prakken's argument game for preferred semantics. We recall that a set of arguments is *ideal* iff it is an admissible set that is a subset of each preferred extension.[3] The *ideal extension* can then be defined as the (unique) maximal ideal set of arguments. It holds that an argument is in the ideal extension iff it is in an admissible set that is not defeated by any admissible set [11]. This means one can first perform the dialogue game for the argument itself, and then the dialogue game against each argument in the thus obtained admissible set to see whether the main argument is in the ideal extension.

The discussion game for stable semantics has so far been described in a relatively informal way, similarly like was done in [19]. It would be interesting to provide a more formalized version of the dialogue game, like for instance was done by Bodenstaff, Prakken and Vreeswijk [3]. Their approach is to use event calculus to formalize the discussion game of [19]. For the stable discussion game, such a formalization would be a topic for future research.

Another topic that is currently left open is that of computational complexity. Although it is known that both credulous preferred and credulous stable are NP-complete problems [9], it still has to be examined how this theoretical complexity

[3] It is also possible to describe an ideal set in terms of labellings. An *ideal labelling* can be defined as an admissible labelling that is a subset of each preferred labelling. It can be proven that $\mathcal{L}$ is an ideal labelling iff $\texttt{in}(\mathcal{L})$ is an ideal set.

of the respective problems relates to actual algorithms that can be defined based on the discussion game described in this paper.

Although the discussion game described in this paper can serve as a basis for the implementation of an algorithm, our main contribution is meant to be conceptual rather than algorithmic. The basic idea of defining a discussion game is to provide a notion of dialectical rationality. That is, the concept of justified information becomes that which can be defended in a rational discussion. The next question, then, is what a rational discussion actually looks like. This question does not have a unique answer; different formalisms implement different ideas about what is a rational discussion. For instance, grounded semantics supports a discussion game in which the burden of proof is higher than, say, the discussion game of preferred semantics. By providing a dialectical semantics of a particular formalism one gains insight in the notion of rationality that the formalism is (implicitly) implementing. In the current paper, we have described how the notion of rational discussion that is implemented by stable semantics differs from the notion of rational discussion that is implemented by preferred semantics. Our aim is that such a comparison is useful by itself, apart from any algorithmic considerations such as efficiency and computational complexity.

References

[1] P. Baroni, M. Giacomin, and G. Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):165–210, 2005.

[2] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.

[3] L. Bodenstaff, H. Prakken, and G. Vreeswijk. On formalising dialogue systems for argumentation in the event calculus. In *Proceedings of the Eleventh International Workshop on Nonmonotonic Reasoning*, pages 374–382, Windermere (UK), 2006.

[4] M.W.A. Caminada. *For the sake of the Argument. Explorations into argument-based reasoning.* Doctoral dissertation Free University Amsterdam, 2004.

[5] M.W.A. Caminada. On the issue of reinstatement in argumentation. In M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, pages 111–123. Springer, 2006. LNAI 4160.

[6] M.W.A. Caminada. Semi-stable semantics. In P.E. Dunne and TJ.M. Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.

[7] M.W.A. Caminada. An algorithm for computing semi-stable semantics. In *Proceedings of the 9th European Conference on Symbolic and Quantitalive Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, number 4724 in Springer Lecture Notes in AI, pages 222–234, Berlin, 2007. Springer Verlag.

[8] C. Cayrol, S. Doutre, and J. Mengin. On Decision Problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation*, 13(3):377–403, 2003.

[9] Y. Dimopoulos and A Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170:209–244, 1996.

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

[11] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[12] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference/Symposium on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[13] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–385, 1991.

[14] J. L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person.* MIT Press, Cambridge, MA, 1995.

[15] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.

[16] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[17] B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J.-J.Ch. Meyer and L.C. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, Utrecht, 1996. Utrecht University.

[18] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[19] G. A. W. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA-00)*, number 1919 in Springer Lecture Notes in AI, pages 239–253, Berlin, 2000. Springer Verlag.