A self-correcting iteration schema for argumentation networks

D. M. Gabbay [a] and O. Rodrigues [b,1]

[a] *Bar Ilan University, Israel, Department of Informatics, King's College London, University of Luxembourg, email: dov.gabbay@kcl.ac.uk*

[b] *King's College London, Department of Informatics, The Strand, London, WC2R 2LS, UK, email: odinaldo.rodrigues@kcl.ac.uk*

Abstract. Given an argumentation network with initial values to the arguments, we look for a numerical algorithm yielding extensions compatible with such initial values. We offer an iteration schema that takes the initial values of the nodes and follows the attack relation producing a sequence of intermediate values that eventually becomes stable leading to an extension in the limit. The schema can be used in abstract as well as in abstract dialectical frameworks (ADFs).

Keywords. argumentation, equational approaches, numerical analysis

1. Orientation and Background

Orientation. The *equational approach* to abstract argumentation frameworks views the network $\langle S,R\rangle$ of an argumentation framework as a generator of equations Eq for functions $f : S \longmapsto U$ (where U is the unit interval $[0,1]$). Any function f which is a solution to the equations is considered a complete numerical Eq extension for the original network and for every argument $X \in S$, we can give the interpretation: if $f(X) = 1$, then X is definitely "in" (**in**); if $f(X) = 0$, then X is definitely "out" (**out**); and a value $f(X) \in (0,1)$ indicates a certain degree of undecidedness about X's acceptance (**und**). We call the values in $\{0,1\}$ *crisp* and the values in $(0,1)$ *undecided*. For a particular choice of equations, namely those of the type $Eq_{\max}$ (to be presented later), the solutions to the equations would exactly correspond to all the complete extensions of the network in Dung's sense. The details of this have been previously worked out in a series of papers [1,9,10,11,12,13].

Using the above as a starting point, we address in this paper the following question. Suppose we have a standard way of writing equations Eq defining the values of nodes according to some intended behaviour of the node interactions in a network $\mathscr{N} = \langle S,R\rangle$. We are then given an initial numerical function $V_0 : S \longmapsto U$ with values for the nodes in $\mathscr{N}$. V_0 may originate as the result of merging networks, voting on the arguments, estimates on the values of arguments, initial guesses in the search for a solution to the equations, etc. Whatever V_0's origins, it may not satisfy the equations, meaning that the original choices for argument acceptance, rejection and undecidedness embedded in it do not correspond to an Eq extension and in the case of $Eq_{\max}$, do not correspond to a complete extension in Dung's sense. We would like to find an extension that closely resembles those choices and for that we need methods for finding the "closest" compatible solution satisfying the equations given the initial values V_0.

[1] Corresponding author.

The reader should note that the same problem arises in traditional three-valued argumentation systems. Caminada and Pigozzi put forward the the concepts of *down-admissibility* and *up-completion* in [6]. The down-admissible set of a set of arguments is its largest admissible subset. A down-admissible set can be turned into a proper extension by "up-completing" it. This is the minimum complete extension that contains it.

In this paper, we propose an iterative procedure that corrects initial values yielding a refinement of Caminada and Pigozzi's results to the more general numerical context (i.e., with varying degrees of undecidedness). We stress the need for this framework because in a wide range of applications, numerical initial values from U will appear at least in an intermediate stage as the result of more complex reasoning about the interactions in the network, or of aggregating multiple networks, etc. It is therefore of practical importance that we know how to deal with such initial values.

The rest of this section will provide the background of our approach. In Section 2 we present the *Gabbay-Rodrigues Iteration Schema* and show how it can be used to calculate extensions given any assignment of initial values to the nodes of an argumentation network. We follow this with a discussion and compare our results with the work of Caminada-Pigozzi in Section 3. Section 4 concludes the paper with a summary of the main results and some future work.

Background. An abstract argumentation framework is a system for reasoning about arguments proposed by Dung [8] and defined in terms of a network $\langle S, R \rangle$, where S is a *finite* non-empty set of arguments and R is a binary relation on S, called the *attack relation*. If $(X,Y) \in R$, we say that the argument X attacks the argument Y and in a directed graph this is depicted with an edge from X to Y. In what follows, $Att(X) = \{Y \in S \mid (Y,X) \in R\}$, i.e., the set of arguments attacking X. If $Att(X) = \varnothing$, then we say that X is a *source node*. For $E \subseteq S$, we write $E \to X$ as a shorthand for $\exists Y \in E$, such that $(Y,X) \in R$.

Given an argumentation framework, one usually wants to reason about the *status* of its arguments, i.e., whether an argument persists or is defeated by other arguments. Source arguments, having no attacks on them, always persist. However, an attack from X to Y may not in itself be sufficient to defeat Y, because X may itself be defeated, and thus the statuses of arguments need to be determined systematically. In Dung's original formulation, this is usually done through *acceptability* conditions for the arguments. A semantics is then defined in terms of *extensions* — subsets of S with special properties. Some important concepts now follow.

A set $E \subseteq S$ is said to be *conflict-free* if for all elements $X, Y \in E$, we have that $(X,Y) \notin R$. Although a conflict-free set only contains elements that do not attack each other, this does not necessarily mean that all arguments in the set are properly supported. Well-supported sets satisfy special *admissibility* criteria. We say that an argument $X \in S$ is *acceptable with respect to E*, if for all $Y \in S$ s.t. $(Y,X) \in R$, there exists an element $Z \in E$ s.t. $(Z,Y) \in R$. A set E is *admissible* if it is conflict-free and all of its elements are acceptable with respect to itself. An admissible set E is a *complete extension* if and only if E contains all arguments which are acceptable with respect to itself. E is called a *preferred extension* of S, if and only if E is maximal with respect to set inclusion amongst all complete extensions of S.

Besides Dung's acceptability semantics, it is also possible to give meaning to the argumentation networks through Caminada's *labelling semantics* [4,5,15]. It uses a labelling function $\lambda : \mathscr{S} \longrightarrow \{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$ satisfying certain conditions. The conditions can be tailored so as to obtain a complete correspondence with Dung's semantics and there is a direct association between extensions and the sets containing the arguments that are labelled **in**. Let dom denote the domain of a function. To facilitate the explanations in the rest of the paper, for a labelling function λ, we define the sets $in(\lambda) = \{X \in \text{dom } \lambda \mid \lambda(X) = \textbf{in}\}$ and $out(\lambda) = \{X \in \text{dom } \lambda \mid \lambda(X) = \textbf{out}\}$ and, analougously, for an assignment $v : S \longmapsto U$ define $in(v) = \{X \in \text{dom } v \mid v(X) = 1\}$ and $out(v) = \{X \in \text{dom } v \mid v(X) = 0\}$.

We say that an argument X is *illegally labelled* **in** by λ, if for some $Y \in Att(X)$, we have that $Y \notin out(\lambda)$; X is illegally labelled **out** by λ, if for all $Y \in Att(X)$, we have that $Y \notin in(\lambda)$; and X is illegally labelled **und** by λ, if either $Att(X) \subseteq out(\lambda)$ or $Att(X) \cap in(\lambda) \neq \varnothing$. Because of the direct correpondence between the labelling semantics and Dung's, we can say that a function λ is *admissible* if λ does not illegally label **in** or **out** any argument in S; and an admissible function λ is *complete* if λ does not illegally label **und** any argument in S. If a labelling function is not admissible, one can recover its maximum "admissible" part by successively turning arguments that are illegally labelled **in** or **out** into **und**. This idea was introduced in [6]:

Definition 1.1 ([6]) *The* down-admissible labelling *of a labelling function λ is the biggest element λ' of the set of all admissible labellings s.t. $in(\lambda') \subseteq in(\lambda)$ and $out(\lambda') \subseteq out(\lambda)$.*

The down-admissible set can now form the basis of a complete extension. The smallest of such extensions is obtained by "up-completing" the set.

Definition 1.2 ([6]) *Let λ be an admissible labelling. The* up-complete labelling of λ *is a complete labelling λ' s.t. $in(\lambda') \supseteq in(\lambda)$ and $out(\lambda') \supseteq out(\lambda)$ and $in(\lambda')$ and $out(\lambda')$ are the smallest sets satisfying these conditions.*

A third approach to the argumentation semantics is Gabbay's *equational approach*, which views an argumentation system $\langle S, R \rangle$ as a mathematical graph generating equations for functions in U [9,10]. Any solution $\mathscr{S}$ to these equations conceptually corresponds to an extension. One equation we can possibly generate is $Eq_{\max}$, where for any node $X \in S$, it numerical value $V(X)$ is defined as $V(X) = 1 - \max_{Y \in Att(X)}\{V(Y)\}$.

Gabbay has shown that in the case of $Eq_{\max}$ the totality of solutions to the system of equations corresponds to the totality of complete extensions in Dung's sense [10]. The equational approach has several advantages, one of which is the possibility of using existing numerical methods to find solutions, and hence extensions.

2. The Gabbay-Rodrigues Iteration Schema

Suppose we are given initial numerical values which do not correspond to any extension using the correspondence between values and labellings presented at the beginning of Section 1. We seek a mechanism that would allow us to find the "best" possible extension corresponding to these initial values in a manner similar to that of Caminada and Pigozzi's. For this we introduce what we call the *Gabbay-Rodrigues Iteration Schema* (for $Eq_{\max}$) which, when applied to a set of initial values, successively corrects these values to produce a complete extension.

Definition 2.1 *Let $\mathscr{N} = \langle S, R\rangle$ be a graph and V_0 assign initial values to the nodes in S. The* Gabbay-Rodrigues Iteration Schema *is defined by the following system of equations T_{GR}, where for each node $X \in S$, iteration $i+1$ is defined in terms of iteration i as follows:*

$$V_{i+1}(X) = (1 - V_i(X)) \cdot \min\{1/2, 1 - \max_{Y \in Att(X)} V_i(Y)\} + V_i(X) \cdot \max\{1/2, 1 - \max_{Y \in Att(X)} V_i(Y)\}$$

We call the system of equations for $\mathscr{N}$ using the schema its *GR system of equations*.

As for Caminada and Pigozzi, the schema iteratively corrects illegal crisp values. When all become correct, we say that the values have become "stable" in the sense that from then on crisp values remain unchanged and undecided values remain undecided (although not necessarily the same).

Definition 2.2 *We say that* the sequence of value assignments $V_0, V_1, \ldots,$ becomes stable at iteration *k, if for all nodes X: 1) If $V_k(X) \in (0,1)$, then $V_{k+1}(X) \in (0,1)$; 2) If $V_k(X) \in \{0,1\}$, then $V_{k+1}(X) = V_k(X)$; and 3) k is the smallest value for which 1. and 2. hold.*

As it turns out, if v is an assigment of initial values to S and the set $in(v)$ corresponds to a complete extension then the sequence of values $V_0 = v$, V_1, V_2,... is stable at the outset (i.e., at iteration 0). If $in(v)$ does not correspond to a complete extension, then at the iteration k where the sequence becomes stable, V_k is the function giving the maximal possible admissible crisp part (i.e., $in(V_k)$ and $out(V_k)$) agreeing with v. This is our numerical counterpart to down-admissibility. However, at iteration k, the undecided values will be the closest possible to the initial values allowing admissibility.

Definition 2.3 (Caminada-Pigozzi Correspondence) *For any labelling function λ define the function $V_\lambda : S \longmapsto \{0, 1/2, 1\}$ as $V_\lambda(X) = 1$ iff $\lambda(X) =$* **in***; $V_\lambda(X) = 0$ iff $\lambda(X) =$* **out** *and $V_\lambda(X) = \frac{1}{2}$ iff $\lambda(X) =$* **und***. For any function $V : S \longmapsto U$ define the labelling function $\lambda_V : S \longmapsto \{$***in**, **out**, **und***$\}$ as $\lambda_V =$* **in** *iff $V(X) = 1$; $\lambda_V(X) =$* **out** *iff $V(X) = 0$ and $\lambda_V(X) =$* **und** *iff $0 < V(X) < 1$.*

Proposition 2.1 *Let λ be a labelling function and V_λ its corresponding Caminada-Pigozzi translation. If the Gabbay-Rodrigues Iteration Schema is employed using V_λ as V_0, then for some value $k \geq 0$, the sequence of values V_0, V_1, ... will become stable and the sets $in(V_k)$ and $out(V_k)$ will correspond to the down-admissible labelling of λ.*

At the limit of the sequence $V_0, V_1, \ldots,$ we reach what we call the *equilibrium values of the nodes*. It can be proved that this limit exists.

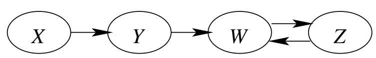
Definition 2.4 *Let $\mathscr{N} = \langle S, R\rangle$ be an argumentation network, T its GR system of equations, and V_0 an assignment of initial values to the nodes in S. The* equilibrium value *of a node $X \in S$ is defined as $V_e(X) = \lim_{i\to\infty} V_i(X)$.*

The nodes with equilibrium value 1 correspond to the up-completion of the set containing the nodes with value 1 at the point the sequence becomes stable.

Theorem 2.1 *Let $\mathscr{N} = \langle S, R\rangle$ be an argumentation network, T its GR system of equations, and V_0 an assignment of initial values to S. If $in(V_0)$ forms a complete extension, then $in(V_e) = in(V_0)$, otherwise $in(V_e)$ is the minimal complete extension containing $in(V_k)$, where k is the point at which the sequence V_0, V_1,... becomes stable.*

3. Discussion and Comparisons with Other Work

Suppose we are given a network such as the one in Figure 1 with some initial values to its nodes. The values may or may not correspond to a complete extension. We can write equations for the network, apply the Gabbay-Rodrigues Iteration Schema and obtain extensions for the network. For the sake of illustration, the table in Figure 1 contains



	X (V_0,V_k,V_e)	Y (V_0,V_k,V_e)	W (V_0,V_k,V_e)	Z (V_0,V_k,V_e)
1.	$(0,3/4,1)$	$(0,1/2,0)$	$(0,0,0)$	$(1,1,1)$
2.	$(0,7/8,1)$	$(1,3/8,0)$	$(1,1/2,1/2)$	$(0,5/8,1/2)$
3.	$(1,1,1)$	$(0,0,0)$	$(1,1,1)$	$(0,0,0)$

Figure 1. Network used in Section 3.

three sets of values **1.**, **2.** and **3.** Each set contains initial (V_0), stable (V_k) and equilibirum values (V_e), calculated using our schema. The corresponding down-admissible labellings and their resulting up-completion according to Caminada-Pigozzi's procedure can be obtained simply by replacing 0 with **out**, 1 with **in** and values in $(0,1)$ with **und**.

Scenario **1.** represents the situation in which the initial values in the cycle $W \leftrightarrow Z$ form an extension and hence the crisp values are preserved by the calculations. We end up with the complete extension $E_{\mathbf{1}} = \{X, Z\}$. Constrast this with case **2.**, in which the initial values of W and Z are 1 and 0, resp. The extension $E = \{X, W\}$ is also complete but is obtained neither by our procedure nor by Caminada-Pigozzi's down-admissible/up-complete construction. This can be explained as follows. The initial illegal value of Y invalidates the initial acceptance of W, turning it into undecided in the calculation of the down-admissible subset. From that point on, the original legal assignments for W and Z can no longer be restored and they both end up as undecided. As a result, we obtain the complete (but not preferred) extension $E_{\mathbf{2}} = \{X\}$. This interference pattern does not occur in case **1.**, because there the undecided value of Y is dominated by Z's stronger value (1/**in**). Since we use max, W's initial value of 0 is retained and hence so is Z's.

If however we start at the outset with a preferred extension, which is also complete by definition, we get as a result unchanged initial values (cf. Theorem 2.1). Caminada-Pigozzi also give the same result because the down-admissible labelling of a labelling yielding a preferred extension is the labelling itself and since that labelling is also complete, then the up-completion does not change anything (case **3.** in the table of Figure 1).

We can suggest an enhanced procedure to improve on the results obtained in case **2.**, which is outlined below. Obviously, the procedure can be adapated to three-valued scenarios and used to obtain a larger complete extension replacing the down-admissible/up-completion steps for the iteration schema. The procedure starts with a set of initial values to the nodes and an empty set of crisp values (*Crisp*).

1. Calculate the equilibrium values of all nodes using the iteration schema.
2. If $\{X \in S \mid V_e(X) \in \{0,1\}\} \subseteq Crisp$, stop. The extension is defined as the set $in(V_e)$. Otherwise, set $Crisp = Crisp \cup \{X \in S \mid V_e(X) \in \{0,1\}\}$ and proceed to step 3.
3. For every X such that $V_e(X) \in \{0,1\}$, set $V_0 = V_e(X)$ and leave $V_0(X)$ as before for the remaining nodes.
4. Repeat from 1.

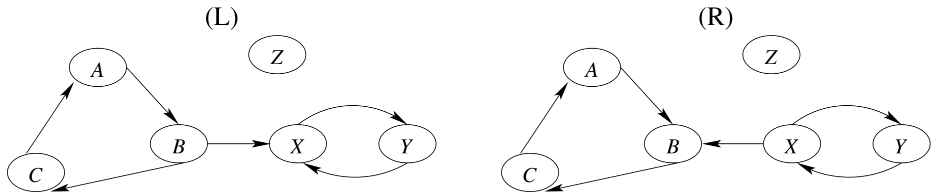
The above procedure is *sound*, since at each run the equilibrium values computed yield a complete extension. Note that re-using some of the original values in step 3. does not affect soundness. If they cannot be used to generate a larger extension, they will just converge to $1/2$. The procedure also *terminates* as long as the original graph S is finite, since a new re-calculation of equilibrium values is invoked only when new crisp values are generated and this is bound by $|S|$.

If we apply the procedure to Case **2.** above, in the first run we will get $V_e(X) = 1$, $V_e(Y) = 0$, $V_e(W) = V_e(Z) = 1/2$. Hence, $Crisp = \{X,Y\}$. We then run it once more, this time with initial value 1 for X and W and 0 for Y and Z. This will stabilise immediately at these values with $Crisp = \{X,Y,W,Z\}$. No further crisp values can be generated, so we stop with extension $\{X,W\}$, which is a preferred extension (see case **3.** above). This result is closer to the original intention, because the acceptance of W (over Z) is preserved.

Comparisons with other work. [6] were the first to look at the problem of finding an extension given an initial labelling of a set of arguments in the way we described it here. The down-admissible labelling is computed by a *contraction sequence* which at each step, turns an illegally labelled argument from **in** or **out** into **und** until no illegal crisp values remain. Our iteration schema produces an equivalent result at the stable point, except that at each iteration it may turn more than one node illegally labelled **in** or **out** into **und** simultaneously and the undecided values we get are more fine-grained (i.e., a value in $(0,1)$). In addition, the application of contractions in [6] is *non-deterministic*: one needs to select an illegally labelled node first and hence there is an implicit cost involved in finding it in the first place. Now given an admissible labelling, the up-completion is done via an *expansion sequence* in [6] which turns illegally undecided nodes into **in** or **out** as appropriate. Our counterpart to the expansion sequence is the calculation of the equilibrium values. Obviously, in a computer implementation we can only approximate these values. In our examples, we stop iterating at the point after which we can no longer guarantee the accuracy of the calculation without introducing rounding errors. This is done in linear time too (see Figure 2). In practice, the limit values can be guessed much earlier as they can be seen to converge to one of the three values 0, $1/2$ and 1. Here one can use the undecided value obtained at the stable point instead of $1/2$.

We stress that neither are we limited to the discrete values **out**, **in** and **und**, nor to the $Eq_{\max}$ equation used in the iteration schema and this allows the application of the schema in the calculation of extensions giving different semantics (see Section 4).

Worked Examples with Cycles. The table in Figure 2 displays initial, stable and equilibrium values (V_0, V_k, V_e) for all nodes in the networks (L) and (R). The last column of the table indicates the iterations at which the stable and equilibrium values were reached (S,E). We set our tolerance as 10^{-19}, the upper bound of the relative error due to rounding in the calculations in our 64-bit machine. Nodes without attackers, such as Z in the graphs and all nodes whose values of all attackers converge to 0 always converge to 1 independently of their initial values. Cases (L) and (R) explore symmetrical scenarios involving cycles. In (L) the odd cycle attacks the even one and in (R) the even cycle attacks the odd one. We start with (L), whose values in the odd cycle will converge to $1/2$ independently of their initial values. The attack of B on X only has an effect on X's value if Y's initial value is not 1. Y's initial value of 1 dominates the effect of the undecidedness of B on X and its value persists as 0. However, if X's initial value is 1 and Y's is 0 (legal values within this cycle), the undecidedness of B will force X to become undecided as well, which in turn also makes Y undecided. As a result, all of the values in the cycles converge to $1/2$. Now let us look at (R) in which the even cycle attacks the odd one. (R1)



	X (V_0, V_k, V_e)	Y (V_0, V_k, V_e)	A (V_0, V_k, V_e)	B (V_0, V_k, V_e)	C (V_0, V_k, V_e)	Z (V_0, V_k, V_e)	(S,E)
L1.	$(0,0,0)$	$(1,1,1)$	$(0, 1/2, 1/2)$	$(1, 0.266, 1/2)$	$(0, 0.562, 1/2)$	$(0, 0.938, 1)$	(3,58)
L2.	$(1, 0.430, 1/2)$	$(0, 0.516, 1/2)$	$(1, 0.516, 1/2)$	$(0, 1/2, 1/2)$	$(0, 0.430, 1/2)$	$(1/2, 0.992, 1)$	(5,58)
R1.	$(1,1,1)$	$(0,0,0)$	$(1, 0.438, 0)$	$(1, 0.062, 0)$	$(0, 0.734, 1)$	$(0, 0.938, 1)$	(3,76)
R2.	$(0,0,0)$	$(1,1,1)$	$(0, 0.562, 1/2)$	$(0, 1/2, 1/2)$	$(1, 0.266, 1/2)$	$(1/2, 0.969, 1)$	(3,58)

Figure 2. Equilibrium and stable values of nodes involved in cycles.

and (R2) contain different initial valid configurations for the even cycle. This time the nodes in the even cycle are independent of external attacks and their original values remain. If the initial value of X is 1, it remains 1 and this in turn breaks the odd cycle. The attacked node B is forced to converge to 0, forcing C to converge to 1 and A to converge to 0 (independently of their initial values). An initial value of 0 for X cannot break the odd cycle and the values of the nodes in that cycle will converge to $1/2$ independently of their initial values (R2).

4. Conclusions and Future Work

This paper investigated aspects concerned with argumentation networks where the arguments are provided with initial values in the unit interval U. We are aware that assigning values to nodes and propagating values through the network has been independently investigated before (e.g., in [2,7]). However, our approach is different because we see a network as a generator for equations whose solutions generalise the concept of extensions of the network.

There are advantages to using equations to calculate extensions in this way as numerical values arise naturally in many applications where argumentation systems are used and the behaviour of the node interactions can be described using equations. In addition, there are many mathematical tools to help find solutions to equations.

Even considering a more restricted three-valued scenario where **und** is any value in $(0,1)$, the values that we obtain at the stable point provide a richer indication of the degree of undecidedness of points in the network.

The equational approach is general enough to be adapted to particular applications. For instance, the arguments themselves may be expressed as some proof in a fuzzy logic and then the initial values can represent the values of the conclusions of the proofs, in the spirit of Prakken's work [14]; or they can be obtained as the result of the merging of several networks, as proposed in [12,13].

The Gabbay-Rodrigues Iteration Schema takes the following *generalised* form:

$$V_{i+1}(X) = (1 - V_i(X)) \cdot \min\{1/2, g(\mathscr{N}(X))\} + V_i(X) \cdot \max\{1/2, g(\mathscr{N}(X))\}$$

In this paper, we considered the special case where g is min and $\mathscr{N}(X)$ is the set of complemented values of the nodes in the "neighbourhood" of X (i.e., the attackers of X).[2] Other operations can be used for argumentation systems, whose relationship with

[2]Note that $1 - \max_{Y \in Att(X)}\{V(Y)\} = \min_{Y \in Att(X)}\{1 - V(Y)\}$.

the schema is being further investigated. One such operation is *product*, which unlike min combines the strength of the attacks on a node. Another interesting possibility is to use the schema for *abstract dialectical frameworks* (ADFs) [3]. ADFs require the specification of a possibly unique type of equation for each node. Consider the ADF with nodes a, b, c and d with $R = \{(a,b),(b,c),(c,c)\}$ and equations $C_a = \top$, $C_b = a$, $C_c = c \wedge b$ and $C_d = \neg d$. The complete models for this ADF are $m_1 = (t,t,u,u)$, $m_2 = (t,t,t,u)$ and $m_3 = (t,t,f,u)$. Our schema converges to m_1 given initial values $(1,1,1/2,1/2)$; to m_2 given initial values $(1,1,1,1)$; and to m_3 given initial values $(0,0,0,0)$.

For the case of min, the values generated at each iteration in the schema eventually "stabilise" by changing illegal crisp values into undecided. This process generalises the notion of down-admissibility proposed in [6], in the sense that undecided values are more fine-grained, and this process can be calculated in time $t \leq |S|$. In the limit of the sequence, some of the undecided values at the stable point will become crisp yielding the up-completion of the down-admissible set and although the remaining undecided values will converge to $1/2$ in the limit, their true degree of undecidedness can be read from their values at the stable point. In practice, only a few iterations are sufficient to indicate what the values will converge to in the limit. If the whole process is iterated it can yield larger and larger complete extensions. These extensions will be as "compatible" as possible with the initial values.

Acknowledgements. We thank M. Caminada and S. Modgil for comments and discussions on the topic of this paper.

References

[1] H. Barringer, D. M. Gabbay, and J. Woods. Temporal dynamics of support and attack networks. In D. Hutter and W. Stephan, editors, *Mechanizing Mathematical Reasoning*, 2005. LNCS, vol. 2605.

[2] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203 – 235, 2001.

[3] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Proceedings of KR'10*, pages 102 – 111. AAAI Press, 2010.

[4] M. Caminada. A labelling approach for ideal and stage semantics. *Argument and Computation*, 2(1):1–21, 2011.

[5] M. Caminada and D. M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[6] M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[7] C. Cayrol and M.-C. Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005.

[8] 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.

[9] D. M. Gabbay. Introducing equational semantics for argumentation networks. *Lecture Notes in Artificial Intelligence*, 6717:19–35, 2011.

[10] D. M. Gabbay. Equational approach to argumentation networks. *Argument and Computation*, 3:87–142, 2012. DOI: 10.1080/19462166.2012.704398.

[11] D. M Gabbay. *Meta-logical Investigations in Argumentation Networks*. College Publications, 2013.

[12] D. M. Gabbay and O. Rodrigues. A equational approach to the merging of argumentation networks. *Journal of Logic and Computation*, 2012.

[13] D. M. Gabbay and O. Rodrigues. A numerical approach to the merging of argumentation networks. In *Proceedings of CLIMA XIII*, pages 195–212. Springer-Verlag, 2012.

[14] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.

[15] Y. Wu and M. Caminada. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.