Equilibrium States in Numerical Argumentation Networks

D. Gabbay  
King’s College London,  
Department of Informatics,  
The Strand,  
London, WC2R 2LS, UK  
dov.gabbay@kcl.ac.uk

O. Rodrigues  
King’s College London,  
Department of Informatics,  
The Strand,  
London, WC2R 2LS, UK  
odinaldo.rodrigues@kcl.ac.uk

August 29, 2014

Abstract

Given an argumentation network with initial values to the arguments, we look for algorithms which can yield extensions compatible with such initial values. We find that the best way of tackling this problem is to offer an iteration formula that takes the initial values and the attack relation and iterates a sequence of intermediate values that eventually converges leading to an extension. The properties surrounding the application of the iteration formula and its connection with other numerical and non-numerical techniques proposed by others are thoroughly investigated in this paper.

1 Orientation and Background

1.1 Orientation

A finite system $\langle S, R \rangle$, with $R$ a binary relation on $S$, can be viewed in many different ways; among them are

1. as an abstract argumentation framework [10], and

2. as a generator of equations [12, 13]
When viewed as an abstract argumentation framework, the basic concepts studied are those of complete extensions (being certain subsets of \( S \)) and different semantics (being sets of complete extensions). When studied as generators of equations, one can generate equations in such a way that the solutions \( f \) to the equations correspond to complete extensions and sets of such solutions correspond to semantics.

This paper offers an iteration formula for finding specific solutions to the equations, responding to initial requirements and compares it with similar demands made in the abstract argumentation case.

We now explain the role iteration formulas play in general in the equational context.

When we have a system of equations designed to model an application area\(^1\) we face two problems: 1) find any solution to the system of equations, which will have a meaning in the application area giving rise to the equations; 2) given boundary conditions and/or other requirements not necessarily mathematical which are meaningful in the application area\(^2\) we would like to find a solution to the system of equations that is compatible/respects the initial conditions/requirements.

These two problems are distinct. The first one of finding any solution is a numerical analysis problem. There are various iteration methods in numerical analysis to find solutions, of which one of the most known is the Newton-Raphson method\(^3\). The second problem is totally different. It calls for an understanding of the requirements coming from the application area and possibly the design of a specialised iteration formula which respects the type of requirements involved.

This paper provides the Gabbay-Rodrigues Iteration Schema, for the case of the equational approach to argumentation, seeking solutions (which we shall see will correspond to a complete extension) respecting as much as possible initial demands and restrictions of what arguments are in or out of the extension. We compare what our iteration schema does with Caminada and Pigozzi’s down-admissible and up-complete constructions\(^4\). Because we are dealing with iteration formulas (involving limits) and we are comparing with set theoretical operations (as in Caminada and Pigozzi’s paper) we have to be detailed and precise and despite it being conceptually clear and simple, the proofs turned out to be mathematically involved, and require some patience from our readers. However, once we establish the properties of our iteration schema, its use and application are straightforward and computationally simple, especially in the context of such tools as MATHEMATICA and others like it. The reader may

---

1. For example, equations of fluid flow in hydrodynamics or equations of particle motion in mechanics, or equations modelling argumentation networks according to the equational approach (to be explained later), or equations modelling a biological system of predator-prey ecology, or some polynomial equation arising in macroeconomics.

2. For example, initial conditions in the case of particle mechanics, or initial size of population in the ecology, or arguments that we would like to be accepted.

3. This method starts with an initial guess of a possible solution and uses various iteration formulae hoping that it will converge to a solution (for an example see how equations can be solved using this method\(^5\)).
wish to just glance at the technical proofs and concentrate on the examples and
discussions. Note the iteration schema idea is very general and applies to other
systems of equations possibly using other iteration formulas.

The actual technical development of the paper will start in Section 2. In
Appendix A we emphasise the distinction between the above two problems with
two detailed examples, the first modelling the dynamics of predator-prey inter-
actions and the second about merging/voting in argumentation networks. We
shall see that the Newton-Raphson iteration method does not work in these
scenarios and that there is the need for a new type of iteration schema. Thus
this paper is not just incremental to the equational approach but constitutes a
serious and necessary conceptual extension.

1.2 Background

Abstract argumentation frameworks were proposed by Dung [10] to reason about
possibly conflicting arguments. They are defined in terms of a tuple \((S, R)\),
where \(S\) is a 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 \(Y\). The tuple \((S, R)\) can be seen as a directed graph, in which an edge
\((X, Y)\) indicates that the direction of attack is from \(X\) to \(Y\) (see Figure 1). In
what follows, we will use the symbol \(\text{Att}(X)\) to denote the set \(\{Y \in S \mid (Y, X) \in R\}\), i.e., the set of arguments attacking an argument \(X\). Following graph theory
convention, if \(X\) has no attackers (i.e., \(\text{Att}(X) = \emptyset\)), we say that \(X\) is a
source node in \((S, R)\). Given a set \(E \subseteq S\), we write \(E \rightarrow X\) as a shorthand for \(\exists Y \in E,\)
such that \((Y, X) \in R\). Furthermore, Following [4], we use \(E^+\) to denote the set
\(\{Y \in S \mid E \rightarrow Y\}\).

\[ \begin{align*}
X \\
\downarrow \\
Y \quad \rightarrow \\
\quad Z
\end{align*} \]

Figure 1: A sample argumentation network.

Given an argumentation network, one usually wants to reason about the status
of its arguments, i.e., whether an argument persists or is defeated by other
arguments. It should be clear that arguments that have 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 be defeated by some argument that attacks it, and
thus one needs an evaluation process to determine the status of all arguments
systematically. In Dung’s original formulation, this was done through an acceptability semantics defining conditions for the acceptability of an argument.
The semantics can be defined in terms of extensions — subsets of \(S\) with special
properties. These subsets are based on two fundamental notions which are
A set $E \subseteq S$ is said to be conflict-free if for all elements $X, Y \in E$, we have that $(X, Y) \notin R$. Intuitively, arguments of a conflict-free set do not attack each other. However, 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 $E \subseteq S$, if for all $Y \in S$, such that $(Y, X) \in R$, there is an element $Z \in E$, such that $(Z, Y) \in R$. A set $E \subseteq S$ 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 admissible extensions of $S$. Similarly, $E$ is called a stable extension of $S$ if and only if $E$ is conflict-free and for every $X \in S \setminus E$, there is an element $Y \in E$, such that $(Y, X) \in R$.

Now consider the argumentation networks (L) and (R) depicted in Figure 2. According to the semantics given above, the network (L) has three extensions $E_0 = \emptyset$, $E_1 = \{X\}$ and $E_2 = \{Y\}$. Both $E_1$ and $E_2$ are preferred and stable extensions. The network (R) only has only one extension, which is empty, and hence this is also its only preferred extension. This extension is however not stable.

Besides Dung’s acceptability semantics, it is also possible to give meaning to these networks through Caminada’s labelling semantics [6, 5] and through Gabbay’s equational approach [12, 13]. These are explained next.

The labelling semantics.

The labelling semantics uses labelling functions $\lambda : S \rightarrow \{\text{in}, \text{out}, \text{und}\}$ satisfying certain conditions tailored so as to obtain a complete correspondence with Dung’s semantics.

The labelling of an argument in disagreement with Dung’s semantics is said to be “illegal”. This is explained further as follows.

**Definition 1.1 (Illegal labelling of an argument [7])** Let $(S, R)$ be an argumentation network and $\lambda$ a labelling function for $S$.

1. An argument $X \in S$ is illegally labelled in by $\lambda$ if $\lambda(X) = \text{in}$ and there exists $Y \in \text{Att}(X)$ such that $\lambda(Y) \neq \text{out}$. 

![Figure 2: Sample argumentation networks.](image-url)
2. An argument \(X \in S\) is illegally labelled \textbf{out} by \(\lambda\) if \(\lambda(X) = \text{out}\) and there is no \(Y \in \text{Att}(X)\) such that \(\lambda(Y) = \text{in}\).

3. An argument \(X \in S\) is illegally labelled \textbf{und} by \(\lambda\) if \(\lambda(X) = \text{und}\) and either for all \(Y \in \text{Att}(X)\), \(\lambda(Y) = \text{out}\) or there exists \(Y \in \text{Att}(X)\), such that \(\lambda(Y) = \text{in}\).

It is possible to have more than one legal labelling function for the same argumentation network. Each labelling function will correspond to an extension in Dung’s semantics. For example, for network (L), we have the three functions \(\lambda_1, \lambda_2\) and \(\lambda_0\) below.

\[
\begin{array}{ccc}
\lambda_1 \leftrightarrow E_1 = \{X\} & \lambda_2 \leftrightarrow E_2 = \{Y\} & \lambda_0 \leftrightarrow E_0 = \emptyset \\
\lambda_1(X) = \text{in} & \lambda_2(X) = \text{out} & \lambda_0(X) = \text{und} \\
\lambda_1(Y) = \text{out} & \lambda_2(Y) = \text{in} & \lambda_0(Y) = \text{und} \\
\end{array}
\]

For the network (R), we have only the function \(\lambda\) such that \(\lambda(X) = \lambda(Y) = \lambda(Z) = \text{und}\). This gives the empty extension.

The equational approach.

The equational approach views an argumentation network \(\langle S, R \rangle\) as a mathematical graph generating equations for functions in the unit interval \([0, 1]\). Any solution \(f\) to these equations conceptually corresponds to an extension. Of course, the end result depends on how the equations are generated and we can get different solutions for different equations. Once the equations are fixed, the totality of the solutions to the system of equations is viewed as the totality of extensions via an appropriate mapping. One equation schema we can possibly use for generating equations is the \(E_{\text{qmax}}\) below, where \(V(X)\) is the value of a node \(X \in S\):

\[
(E_{\text{qmax}}) \quad V(X) = 1 - \max_{Y_i \in \text{Att}(X)} \{V(Y_i)\}
\]

Another possibility is \(E_{\text{inv}}\):

\[
(E_{\text{inv}}) \quad V(X) = \prod_{Y_i \in \text{Att}(X)} (1 - V(Y_i))
\]

It is easy to see that according to \(E_{\text{qmax}}\) the value of any source argument will be 1 (since they have no attackers) and the value of any argument with an attacker with value 1 will be 0. The situation is more complex with nodes participating in cycles. Consider the network (L) again, with equations

\[
\begin{align*}
V(X) &= 1 - V(Y) \\
V(Y) &= 1 - V(X)
\end{align*}
\]

If values are taken from the unit interval, this system of equations will accept any solution \(V\) such that \(V(X) + V(Y) = 1\). We can divide these solutions between three classes: \(V^1(X) = 1, V^1(Y) = 0; V^2(X) = 0, V^2(Y) = 1\) and \(0 < V^0(X) < 1, 0 < V^0(Y) < 1\) with \(V^0(X) + V^0(Y) = 1\). These again correspond to the three extensions \(E_1, E_2\) and \(E_0\) given before.

5
In fact, Gabbay has shown that in the case of $Eq_{\text{max}}$ the totality of solutions to the system of equations corresponds to the totality of extensions in Dung’s sense. The correspondence is best explained in terms of the labelling semantics, using the following correspondence:

\[
V(X) = 1 \quad : \quad \lambda(X) = \text{in} \\
V(X) = 0 \quad : \quad \lambda(X) = \text{out} \\
0 < V(X) < 1 \quad : \quad \lambda(X) = \text{und}
\]

The advantage of the equational approach is that it allows us to think of an argumentation network as a numeric system in which nodes are given certain values depending on specific rules governing their interaction with their neighbours. A rule may for instance require the value of a node to be 0 if the value of any attacking node is 1. Another rule may force the value of a node to be 1 if it has no attacking nodes. The schema $Eq_{\text{max}}$ embeds these rules, and they agree with Dung’s semantics. A solution to the system of equations is any combination of values of nodes satisfying the equations. Of course, since the node values are no longer discrete we have more freedom to design rules which are appropriate for a given application. Part of the objective of this paper is to explore the nature of these rules.

We start by generalising some concepts a bit further. Consider the network in Figure 3 in which $\text{Att}(X) = \{Y_1, Y_2, \ldots, Y_k\}$. To agree with Dung’s semantics, if the value of any attacker of $X$ is 1, we want the value of $X$ to be 0. If all of the attackers of $X$ have value 0, we want the value of $X$ to be 1. For any other combination of values of the attackers we want the value of $X$ to be anything other than 0 or 1. So within the traditional semantics but taking the extended set of values of the unit interval, we can think of a single attack by a node with value $v$ as the order-reversing operation which returns the value $1 - v$. This is a kind of negation. Since a node can have multiple attacks, we also need an operation to combine the values of the attackers. We can think of this as a type of conjunction, which numerically can be obtained through several operations. For instance, in fuzzy logic, the standard semantics of (weak) conjunction is given by the operation $\min$.

\[X \lor Y \lor Y_1 \lor Y_2 \lor \ldots \lor Y_k\]

Figure 3: Multiple attacks on a node.

4If we make und equals $\frac{1}{2}$, then an attack by a single undecided node will have value $\frac{1}{2}$. 
Therefore, the value of a node $X$ can be defined as

$$V(X) = \min_{Y \in \text{Att}(X)} \{1 - V(Y)\}$$

which is equivalent to

$$V(X) = 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\}$$

obtained by our now familiar schema $E_{q_{\text{max}}}$. Note that the conjunction operation in the schema $E_{q_{\text{max}}}$ is product. The operations min and product are two examples of t-norms. They are two instances of functions that are particularly suitable for argumentation semantics. The following definition elaborates on this further.

**Definition 1.2** A function $g : U^2 \rightarrow U$ is argumentation-friendly if $g$ satisfies the following conditions.

(T1) $g(x, y) = g(y, x)$

(T2) $g(x, g(y, z)) = g(g(x, y), z)$

(T3) $g(0, x) = 0$

(T4) $g(1, x) = x$

(T5) $g$ is continuous as a multi-variable function.

We can extend $g$ to $2^U$ as follows. $g(\emptyset) = 1; g(\{x\}) = x$; and $g(\{x\} \cup \Delta) = g(x, g(\Delta))$. It is easy to see that if $0 \in \Delta$, then $g(\Delta) = 0$ and that $g(\{1\}) = 1$.

Later on, we will see that argumentation-friendly functions will be used both to calculate aggregation of attacks as well as for combining the value of attacks with initial values. However, as we mentioned attack is a type of negation and hence when operating on the attack of a node with value $v$, we will consider the complement of $v$ to 1, i.e., $(1 - v)$.

Notice that t-norms satisfy conditions (T1)–(T4) above.

**Definition 1.3** For any assignment of values $v : S \rightarrow U$ define the sets $\text{in}(v) = \{X \in \text{dom} v \mid v(X) = 1\}$ and $\text{out}(v) = \{X \in \text{dom} v \mid v(X) = 0\}$.

**Theorem 1.1** Let $\mathcal{N} = (S, R)$ be a network, $g$ an argumentation-friendly function, and $T$ a system of equations written for $\mathcal{N}$, where for each node $X$, $V(X) = g_{Y \in \text{Att}(X)} \{1 - V(Y)\}$. Take any solution $V$ to $T$, it follows that $\text{in}(V)$ is a complete extension.

**Proof.** $\text{in}(V)$ is conflict-free, for if you take $X \in \text{in}(V)$, then we have that $V(X) = 1$, but since $V(X) = g_{Y \in \text{Att}(X)} \{1 - V(Y)\}$, then $(1 - V(Y)) = 1$, for all $Y \in \text{Att}(X)$ and therefore if $(Y, X) \in R$, then $V(Y) = 0$ and then $Y \notin \text{in}(V)$.

\[\text{In fact, this condition is only needed to guarantee the existence of solutions to the equations.}\]
in(V). Similarly, if (X, Y) ∈ R and X ∈ in(V), then V(Y) = g(\{0, \ldots\}) = 0 and then Y \not\in in(V). In fact, V(Y) = 0, for all Y such that Y ∈ in(V)⁺. Therefore, if Att(Z) \subseteq in(V)⁺, then for all W ∈ Att(Z), V(W) = 0 and hence V(Z) = g(\{1\}) = 1, and hence Z \in in(V). This shows that in(V) is a complete extension.

**Theorem 1.2** Let \( N = (S, R) \) be a network, \( g \) an argumentation-friendly function, and \( T \) a system of equations written for \( N \), where for each node \( X \), \( V(X) = g_{Y \in Att(X)}[1 - V(Y)] \). Then for every preferred extension \( E_N \) of \( N \), there exists a solution \( V \) to \( T \) such that

1. If \( X \in E_N \), then \( V(X) = 1 \)
2. If \( E_N \nrightarrow X \), then \( V(X) = 0 \)
3. If \( X \not\in E_N \) and \( E_N \nrightarrow X \), then \( 0 < V(X) < 1 \)

**Proof.** Generate a new system of equations \( T' \) by replacing the original equations in \( T \) by equations in which the variables of all nodes satisfying conditions (C1) and (C2) above have been replaced with the value 0 or 1 as given. It is easy to see that we will be left with the following types of equations.

1. \( 1 = g(\Delta_1) \)
2. \( 0 = g(\Delta_2) \)
3. \( V(X_i) = g(\Delta_3) \) (for \( 0 < V(X_i) < 1 \))

Equations of type (1) refer to nodes \( X \in E_N \). Since \( E_N \) is a complete extension and hence admissible, we must have that i) either \( Att(X) = \Delta_1 = \emptyset \), but then \( g(\Delta_1) = 1 \); or ii) \( E \nrightarrow Y \), for all \( Y \in Att(X) \), for if \( E \nrightarrow Y \), for some \( Y \in Att(X) \), we would have that \( E_N \) would not be admissible. Therefore, by (C2), we have that for all \( Y \in Att(X) \), \( V(Y) = 0 \) and hence \( 1 - V(Y) = 1 \). Therefore, \( \Delta_1 = \{1\} \). Hence, all equations of type (1) are satisfied.

Equations of type (2) refer to a node \( X \) such that \( E_N \nrightarrow X \). Let \( Y \in E_N \), be such that \( (Y, X) \in R \). Since \( Y \in E_N \), then \( V(Y) = 1 \), and hence \( 1 - V(Y) = 0 \). Therefore, \( 0 \in \Delta_2 \), and hence \( g(\Delta_2) = 0 \), satisfying those equations.

Equations of type (3) have a value \( V(X_i) \) on the left-hand side and a set of values \( \Delta_3 \) on the right-hand side. The values in \( \Delta_3 \) are all greater than 0, for if one of them is 0, then the associated node would have value 1, and hence the value on the left-hand side of the equality would have to be 0 (by (C2)). Analogously, \( \Delta_3 \neq \{1\} \), because this would reduce to case (1). Therefore, \( \Delta_3 \) must include at least one value \( v \) such that \( 0 < v < 1 \) and possibly the value 1. A solution will exist because of Brouwer’s fixed-point theorem and this solution cannot give value 0 or 1 to any of the remaining unsolved variables in the equation, because if it did, by Theorem [1.1] this solution would give rise to a complete extension \( in(V) \) such that \( in(V) \supseteq E_N \), and therefore \( E_N \) would not be preferred.
The condition of preferred extension of the Theorem 1.2 is necessary, as shown in the example below.

**Example 1.1** Consider the complete extension \( E = \{X\} \) of the network below. 

\( E \) is not preferred, since \( E \) is a proper subset of \( \{X,W\} \).

\[
\begin{align*}
  V(X) &= 1 - V(Y) \quad (1) \\
  V(Y) &= 1 - V(X) \quad (2) \\
  V(W) &= 1 - V(Z) \quad (3) \\
  V(Z) &= g(\{1 - V(W), 1 - V(Z)\}) \quad (4)
\end{align*}
\]

Since \( V(X) = 1 \), we get that \( V(Y) = 0 \) and these values satisfy equations (1) and (2) above. However, replacing (3) in (4) gives us

\[
V(Z) = g(V(Z), 1 - V(Z))
\]

If \( g \) is product, this gives us \( V(Z) = V(Z) \cdot (1 - V(Z)) \), and hence \( 1 = 1 - V(Z) \) \( \therefore \) \( V(Z) = 0 \), and hence \( V(W) = 1 \), and therefore no solution corresponding to \( E \) using \( g \) exists. Note that the two semi-stable extensions, i.e., \( \{X,W\} \) and \( \{Y,W\} \) include \( W \). No extension can include \( Z \).

However, with \( g \) as \( \min \), we have that (4) becomes

\[
V(Z) = \min(\{1 - V(W), 1 - V(Z)\})
\]

and for this set of equations, the values \( V(X) = 1 \), \( V(Y) = 0 \), \( V(W) = V(Z) = \frac{1}{2} \) form a solution corresponding to \( E \).

The loop in the example above is quite elucidating. Let us analyse it in some more detail.

**Example 1.2** Consider the network with a single self-referencing loop below.

\[
\begin{align*}
  V(X) &= g(\{1 - V(X)\})
\end{align*}
\]

Notice that \( g(\{1 - V(X)\}) = 1 - V(X) \) and hence we have that \( V(X) = 1 - V(X) \) \( \therefore \) \( V(X) = \frac{1}{2} \), whatever the function \( g \) is, as long as it satisfies (T1)–(T5).
Note that min satisfies (T1)–(T4). As a result, we have that:

**Corollary 1.1** Let \( N = \langle S, R \rangle \) be a network and \( T \) a system of equations written for \( N \), where for each node \( X \), \( V(X) = \min_{Y \in \text{Att}(X)} \{1 - V(Y)\} \). Take any solution \( V \) to \( T \). It follows that \( \text{in}(V) \) is a complete extension.

In other words, any solution to the system of equations defined in terms of \( E_{\text{eqmax}} \) can be translated into a complete extension simply by defining that extension as the set containing the nodes whose solution values are 1. Obviously, different solutions will give rise to different extensions.

**Proposition 1.1** Let \( N = \langle S, R \rangle \) be a network and \( T \) a system of equations written for \( N \), where for each node \( X \), \( V(X) = \min_{Y \in \text{Att}(X)} \{1 - Y\} \). Then for every complete extension \( E \) of \( N \), there exists a solution \( V \) to \( T \) satisfying:

\[
\begin{align*}
(C1) & \text{ If } X \in E, \text{ then } V(X) = 1. \\
(C2) & \text{ If } E \rightarrow X, \text{ then } V(X) = 0. \\
(C3) & \text{ If } X \notin E \text{ and } E \not\rightarrow X, \text{ then } 0 < V(X) < 1.
\end{align*}
\]

**Proof.** Let \( E \) be a complete extension. Consider the following assignment of values to the nodes in \( S \):

- if \( X \in E \), then \( V(X) = 1 \)
- if \( E \rightarrow X \), then \( V(X) = 0 \)
- \( V(X) = \frac{1}{2} \), otherwise

We now show that the values above form a solution to the system of equations \( T \). As in Theorem 1.2, replacing the above values in the original system of equations will reduce them to the following types.

\[
\begin{align*}
(1) & \quad 1 = \min(\Delta_1) \\
(2) & \quad 0 = \min(\Delta_2) \\
(3) & \quad \frac{1}{2} = \min(\Delta_3)
\end{align*}
\]

We have seen that \( \Delta_1 = \{1\} \) and since \( 1 = \min(\{1\}) \), (1) is satisfied. Similarly, \( 0 \in \Delta_2 \) and since \( \min(\{0, \ldots\}) = 0 \), so is (2). Notice that the image of \( V \) is \( \{0, 1/2, 1\} \). All values in \( \Delta_3 \) are greater than 0, but at least one of them is \( \frac{1}{2} \), therefore \( \min(\Delta_3) = \frac{1}{2} \), and hence the above assignment solves the equations.

So far, we have shown the basics of the equational numerical approach to abstract argumentation frameworks. In the next section we consider two additional developments that follow naturally. Firstly, we know that solutions do exist to the system of equations, but can we find them using some numerical
method? For example, by applying iterations given some initial guess?\textsuperscript{6} Secondly, we would like to apply our methodology to questions of merging, voting, or any other application where a set of initial values emerges and needs to be transformed to the “closest” extension. How can we do that? The following section provides a method to answer these questions.

2 The Gabbay-Rodrigues Iteration Schema

Suppose we are given initial values which do not correspond to any extension in the way that we presented them in the previous section. We seek a mechanism that would allow us to find the “best” possible extension corresponding to these initial values.

Consider the equation $Eq_{\text{max}}$:

$V(X) = 1 - \max_{Y_i \in \text{Att}(X)} \{V(Y_i)\}$

It is satisfied when the value of the node $X$ is legal in Caminada and Pigozzi’s terminology\textsuperscript{7}. That is, if the value of $X$ is 1 and the value of all of $X$’s attackers are 0; or if the value of $X$ is 0 and at least one of $X$’s attackers has value 1.

If we aim to correct the values of the nodes in a network iteratively, we need a mechanism that leaves legal in or out node values intact, changing illegal in or out values into und. To make a distinction between these classes, we will call the values in $\{0, 1\}$ crisp and the values in $(0, 1)$ undecided.

Now consider the following averaging function:

$$(1 - X) \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\} + X \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\}$$

For legal assignments of values, we have three cases to consider:

- $X$ is legally in. In this case $X = 1$ and all of its attackers have value 0. We want the value of $X$ to remain 1. We have that:

  $$(1 - X) \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\} + X \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\} =$$

  $$1 \cdot \max\{\frac{1}{2}, 1\} =$$

  $$= 1$$

- $X$ is legally out. In this case $X = 0$ and at least one of its attackers has value 1. We want the value of $X$ to remain 0. We have that:

  $$1 \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\} + X \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\} =$$

  $$1 \cdot \min\{\frac{1}{2}, 0\} + 0 \cdot \max\{\frac{1}{2}, 0\} =$$

  $$= 0$$

\textsuperscript{6}As can be done to find the square root of numbers using Newton-Raphson’s method.

11
• $X$ is legally **und**. In this case $0 < X < 1$, none of its attackers has value 1 and at least one of its attackers has value greater than 0. This means that $0 < \max_{Y \in \text{Att}(X)} Y < 1$ and therefore $0 < (1 - \max_{Y \in \text{Att}(X)} Y) < 1$. Let $\alpha_1 = \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\}$ and $\alpha_2 = \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\}$. It follows that $0 < \alpha_1 < 1$ and $0 < \alpha_2 < 1$. We want the value of $X$ to remain undecided, although we are prepared to accept changes to its initial value as long as its final value remains in the interval $(0, 1)$. We have that:

$$(1 - X) \cdot \min\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\right\} + X \cdot \max\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\right\} =

(1 - X) \cdot \alpha_1 + X \cdot \alpha_2 =

\alpha_1 - X \cdot \alpha_1 + X \cdot \alpha_2 =

\alpha_1 - X \cdot (\alpha_1 - \alpha_2) = \kappa$$

Notice that $\alpha_1 \leq \frac{1}{2}$ and $\alpha_2 \leq \frac{1}{2}$. If $\alpha_1 = \alpha_2$, then $\kappa = \alpha_1$ and hence $0 < \kappa \leq \frac{1}{2}$. If $\alpha_1 < \alpha_2$, then $-\frac{1}{4} < (\alpha_1 - \alpha_2) < 0$, then $\alpha_1 < \kappa < \alpha_1 + \frac{1}{4}$, and hence $0 < \kappa < \frac{3}{4}$, and therefore the value of $X$ remains in $(0, 1)$.

In summary, we follow Caminada and Pigozzi [7]: we leave legal crisp values intact and legal undecided values within the undecided range. Later we will see that we also turn illegal crisp values into undecided and on a second phase we “correct” illegal undecided values. Our approach does this iteratively though, through what we call the Gabbay-Rodrigues Iteration Schema, introduced below:

**Definition 2.1** Let $\mathcal{N} = (S, R)$ be an argumentation network and $V_0$ be an assignment of values to the nodes in $S$. The Gabbay-Rodrigues Iteration Schema is defined by the following system of equations $T$, where for each node $X \in S$, the value $V_{i+1}(X)$ is defined in terms of the values of the nodes in $V_i$ as follows:

$$T \quad V_{i+1}(X) = (1 - V_i(X)) \cdot \min\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\right\} +

V_i(X) \cdot \max\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\right\}$$

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

We ask whether we can regard the iteration schema above as an **equation** schema as in the previous section, i.e.,

$$X = (1 - X) \cdot \min\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\right\} + X \cdot \max\left\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y\right\}$$

Note that the above equation schema is implicit as $X$ appears on both sides. Therefore, the system of equations arising from a network $(S, R)$ is not guaranteed by Brower’s theorem to have a solution. However, if we can show that the distance between the values of every node $X$ between successive iterations decreases, there will be a solution, as the iteration values will converge. To further
clarify this point, let us take an equation written with an argumentation-friendly function $g$ for a node $X$ in terms of its attackers. The equation would be

$$X = g(\cup_{Y \in \text{Att}(X)} \{1 - Y\})$$

It is clear that if one of the attackers of $X$ is 1, the value of $X$ solves to 0, and if all the attackers of $X$ are 0, the value of $X$ will solve to 1. This follows from the properties (T1)–(T5) of an argumentation-friendly function. Now let us compare and see what happens when we use the formula above. If the value of one of the attackers of $X$ is 1, the first component of the sum will be 0, whereas the second component will be $\frac{1}{2}$, because the equation is implicit, we have the equation

$$X = \frac{X}{2}$$

which solves to $X = 0$, which is correct. If the values of all attackers of $X$ are 0, then we get the equation

$$X = \frac{(1 - X)}{2} + X$$

which solves to $X = 1$, which again gives a correct result. Otherwise, assume that the values of all attackers are either 0 or $\frac{1}{2}$, with at least one of them being $\frac{1}{2}$. We get the equation

$$X = \frac{(1 - X)}{2} + \frac{X}{2}$$

which again solves to the correct value of $X = \frac{1}{2}$. By correct we mean that the results are exactly compatible with the Caminada labelling mentioned in Section 1, where $X = 1$ means $X$ is in, $X = 0$ means $X$ is out and $X = \frac{1}{2}$ means $X$ is und.

Therefore, the Gabbay-Rodrigues equation remains faithful to Dung’s semantics just as $Eq_{\text{max}}$ does. Its advantage over $Eq_{\text{max}}$ is that it can be used iteratively as we will show in the rest of this section.

We start by showing some properties of the schema. The first one ensures that the values of all nodes remain in the unit interval in all iterations.

**Proposition 2.1** Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network and $V_0 : S \mapsto U$ an assignment of initial values to the nodes in $S$. Let each assignment $V_i$, $i > 0$, be calculated by the Gabbay-Rodrigues Iteration Schema for $\mathcal{N}$. It follows that $V_i(X) \in U$, for all $i \geq 0$ and all $X \in S$.

**Proof.** The base of the induction is the initial value assignment that holds trivially. The induction step is proven by looking at the maximum and minimum values that the nodes can take and showing that the sum in the iterated schema is always a number in $U$. Now, suppose that indeed for all nodes $X \in S$, $0 \leq V_k(X) \leq 1$, for a given iteration $k$. Pick any node $X$. It follows that

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

13
So we have that $V_{k+1}(X) = (1 - \alpha) \cdot x + \alpha \cdot y$, where $0 \leq \alpha \leq 1$, $0 \leq (1 - \alpha) \leq 1$, $0 \leq x \leq 1/2$, and $1/2 \leq y \leq 1$.

The lowest value for $V_{k+1}(X)$ is obtained with the lowest values for $x$ and $y$, when we get that $V_{k+1}(X) = \frac{1}{2}$. If $\alpha = 0$, then $V_{k+1}(X) = 0 \geq 0$. If $\alpha = 1$, then we get $V_{k+1}(X) = 1/2 \leq 1$. The highest value for $V_{k+1}(X)$ is obtained with the highest values for $x$ and $y$, when we get that $V_{k+1}(X) = \frac{(1 - \alpha)}{2} + \alpha$. If $\alpha = 0$, then $V_{k+1}(X) = 1/2 \leq 1$. If $\alpha = 1$, then we get $V_{k+1}(X) = 1 \leq 1$. In all cases, $0 \leq V_{k+1}(X) \leq 1$.

We now show that a given “legal” set of initial values for the nodes in $S$ satisfies the equations and hence the values remain unchanged.

**Proposition 2.2** Let $\mathcal{N} = \langle S, R \rangle$ be a network and $T$ its GR system of equations. Then for every complete extension $E$ of $\mathcal{N}$ and all $X \in S$, if $V_0$ is defined using $E$ by the clauses (C1)–(C3) below, we have that $V_1(X) = V_0(X)$.

(C1) If $X \in E$, then $V_0(X) = 1$

(C2) If $E \to X$, then $V_0(X) = 0$

(C3) If $X \not\in E$ and $E \not\to X$, then $V_0(X) = \frac{1}{2}$

**Proof.** Let $E$ be a complete extension and suppose $V_0(1)$. Then $X \in E$ and hence, i) either $\text{Att}(X) = \emptyset$, or ii) for all $Y \in \text{Att}(X)$, $E \to Y$ (since $E$ is admissible). As a result, $1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} = 1$, and hence we have that

$$V_1(X) = \max\{\frac{1}{2}, 1\} = 1 = V_0(X).$$

If on the other hand, $V_0(X) = 0$, then $E \to X$. Therefore, there is some $Y \in \text{Att}(X)$, such that $Y \in E$ and hence $V_0(Y) = 1$. It follows that

$$V_1(X) = \min\{\frac{1}{2}, 1 - 1\} = 0 = V_0(X).$$

Finally, if $V_0(X) = \frac{1}{2}$, then $X \not\in E$ and $E \not\to X$. We must have that for all $Y \in \text{Att}(X)$, $V_0(Y) < 1$ (otherwise, we would have that $E \to X$). We must also have that for some $Y \in \text{Att}(X)$, $V_0(Y) > 0$, otherwise $E$ would defend $X$ and since it is complete $X \in E$, but then $V_0(X) = 1$. Therefore, $1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} = \frac{1}{2}$, and hence we have that

$$V_1(X) = \frac{1}{2} \cdot \min\{\frac{1}{2}, \frac{1}{2}\} + \frac{1}{2} \cdot \max\{\frac{1}{2}, \frac{1}{2}\} = \frac{1}{4} + \frac{1}{4} = \frac{1}{2} = V_0(X).$$

Obviously, if for all nodes $X$, $V_1(X) = V_0(X)$ as above, then for all nodes $X$, $V_{i+1}(X) = V_i(X)$, for all $i \geq 0$.

Furthermore, crisp values do not “swap” between each other and undecided values do not become crisp:
Theorem 2.1  Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network, $T$ a system of equations for $\mathcal{N}$ using the Gabbay-Rodrigues Iteration Schema, and $V_0 : S \mapsto U$ an assignment of initial values to the nodes in $S$. Let $V_0, V_1, V_2,$ be a sequence of value assignments where each $V_i$, $i > 0$, is generated by $T$. Then the following properties hold for all $X \in S$ and for all $k \geq 0$

1. If $V_k(X) = 0$, then $V_{k+1}(X) \neq 1$.
2. If $V_k(X) = 1$, then $V_{k+1}(X) \neq 0$.
3. If $0 < V_k(X) < 1$, then $0 < V_{k+1}(X) < 1$.

Proof.

1. Suppose $V_k(X) = 0$, then $V_{k+1}(X) = \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\} \leq \frac{1}{2}$.
2. Suppose $V_k(X) = 1$, then $V_{k+1}(X) = \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\} \geq \frac{1}{2}$.
3. Suppose $0 < V_k(X) < 1$. We first show that $V_{k+1}(X) \neq 0$. Note that $0 < (1 - V_k(X)) < 1$. Suppose that $V_{k+1}(X) = 0$. We have that

$$0 = (1 - V_k(X)) \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\} + V_k(X) \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\}$$

Therefore, $\min\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\} = 0$, but then $\max\{\frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y)\} = \frac{1}{2}$, and since $V_k(X) > 0$, so is $\frac{V_k(X)}{2}$, a contradiction.

Since we start with values in $U$, Proposition 2.1 gives us that $V_{k+1}(X) \leq 1$, for all $X \in S$. We therefore only need to show that $V_{k+1}(X) \neq 1$. Again we have that $V_{k+1}(X) = (1 - \alpha) \cdot x + \alpha \cdot y$, where

$$0 < \alpha < 1$$

$$0 < (1 - \alpha) < 1$$

$$0 \leq x \leq \frac{1}{2}$$

$$\frac{1}{2} \leq y \leq 1$$

Suppose $V_{k+1}(X) = 1$. It follows that

$$(1 - \alpha) \cdot x + \alpha \cdot y = 1$$

$$x - \alpha \cdot x + \alpha \cdot y = 1$$

$$\alpha (y - x) = (1 - x)$$

$$\alpha = \frac{1 - x}{y - x}$$

Since $\alpha < 1$, we have that $1 - x < y - x$, and hence $y > 1$, a contradiction.

15
The above theorem shows that any changes between iterations can only generate new values for nodes in the interval \((0,1)\), i.e., successive iterations can only turn crisp values into undecided. Therefore, the sets of nodes with crisp values can only decrease throughout the iterations:

**Corollary 2.1** Let \(\mathcal{N} = (S,R)\) be an argumentation network, \(V_0 : S \rightarrow U\) an initial assignment of values to the nodes in \(S\) and \(T\) its GR system of equations. It follows that for all \(0 \leq i \leq j\), \(\text{in}(V_j) \subseteq \text{in}(V_i)\) and \(\text{out}(V_j) \subseteq \text{out}(V_i)\).

The situation in the limit of the sequence of values is more complex and we will deal with it later. If between two successive iterations there are no changes in the crisp values, then these values “stabilise”:

**Theorem 2.2** Let \(\mathcal{N} = (S,R)\) be a network, \(T\) its GR system of equations, and \(V_0\) an initial assignment of values to the nodes in \(S\). Let \(V_0, V_1, V_2, \ldots\) be a sequence of value assignments where each \(V_i, i > 0\), is generated by \(T\). Assume that for some iteration \(i\) and all nodes \(X \in S\) such that \(V_i(X) \in \{0,1\}\), we have that \(V_{i+1}(X) = V_i(X)\), then for all \(j \geq 1\), \(V_{i+j}(X) = V_i(X)\).

**Proof.** Assume that \(V_i(X) \in \{0,1\}\) for some node \(X\). There are two cases to consider.

**Case 1:** \(V_i(X) = 0\). By assumption, we have that \(V_{i+1}(X) = 0\). We show that \(V_{i+2}(X) = 0\). If \(V_{i+1}(X) = 0\), we have that

\[
V_{i+1}(X) = (1 - V_i(X)) \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} \right\} + V_i(X) \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} \right\}
\]

So, \(\max_{Y \in \text{Att}(X)} \{V(Y)\} = 1\) and hence for some \(Y \in \text{Att}(X)\), \(V_i(Y) = 1\). By assumption \(V_{i+1}(Y) = 1\) and hence \(\max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} = 1\). Therefore,

\[
V_{i+2}(X) = \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} \right\} = 0
\]

**Case 2:** \(V_i(X) = 1\). By assumption, we have that \(V_{i+1}(X) = 1\). We show that \(V_{i+2}(X) = 1\). If \(V_{i+1}(X) = 1\), we have that

\[
V_{i+1}(X) = (1 - V_i(X)) \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} \right\} + V_i(X) \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} \right\}
\]

So, \(\max_{Y \in \text{Att}(X)} \{V(Y)\} = 1\) and hence for some \(Y \in \text{Att}(X)\), \(V_i(Y) = 1\). By assumption \(V_{i+1}(Y) = 1\) and hence \(\max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} = 1\). Therefore,

\[
V_{i+2}(X) = \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} \right\} = 0
\]
So, \( \max_{Y \in \text{Att}(X)} \{V_i(Y)\} = 0 \), and hence for all \( Y \in \text{Att}(X) \), \( V_i(Y) = 0 \). By assumption, \( \max_{Y \in \text{Att}(X)} \{V_i(Y)\} = 0 \), and hence

\[
V_{i+2}(X) = \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} \right\} = 1
\]

**Definition 2.2** Let \( N = \langle S, R \rangle \) be an argumentation network and \( V_0 : S \rightarrow U \) an assignment of initial values to the nodes in \( S \). A sequence of assignments \( V_i : S \rightarrow U \) where each \( i > 0 \) is generated by the Gabbay-Rodrigues Iteration Schema for \( N \) becomes stable at iteration \( k \), if for all nodes \( X \in S \) we have that

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. above hold.

**Corollary 2.2** Consider a sequence of value assignments \( V_0, V_1, V_2, \ldots \) as described in Theorem 2.2. If the sequence becomes stable at iteration \( k \), then the sequence remains stable for all iterations \( k + j \), \( j \geq 0 \).

**Proof.** The first stability condition in Definition 2.2 follows from Theorem 2.1 and the second condition follows from Theorem 2.2.

**Corollary 2.3** Let \( N = \langle S, R \rangle \) be an argumentation network, \( V_0 : S \rightarrow U \) an assignment of initial values to the nodes in \( S \) and \( T \) its GR system of equations. The following hold:

1. If the sequence of value assignments is not stable at iteration \( k \), then there exists \( X \in S \), such that \( V_k(X) \in \{0, 1\} \) and \( V_{k+1}(X) \in (0, 1) \).
2. Let \( |S| = n \). Then, the sequence is stable for some \( k \leq n \).

**Proof.** [1] follows from Theorem 2.1. For \[2\], notice that each iteration \( i \) which is not stable causes at least one node to change value from \( \{0, 1\} \) into \( (0, 1) \). Theorem 2.1 states that all values in \( (0, 1) \) remain in \( (0, 1) \). Since \( S \) is finite, there are only finitely many nodes that can change from \( \{0, 1\} \) into \( (0, 1) \) and the number of iterations in which this can happen is bounded by \( |S| \).

Corollary 2.3 shows that for some value \( 0 \leq k \leq |S| \), the sequence of value assignments \( V_0(X), V_1(X), V_2(X), \ldots \) eventually becomes stable. That is, there exists \( k \geq 0 \), such that for all \( j \geq 0 \) and all nodes \( X \)

- if \( V_k(X) = 0 \), then \( V_{k+j} = 0 \);
- if \( V_k(X) = 1 \), then \( V_{k+j} = 1 \); and
- if \( V_k(X) \in (0, 1) \), then \( V_{k+j} \in (0, 1) \).
Remark 2.1 Given an argumentation-friendly function $g$, we can define the Gabbay-Rodrigues Iteration Schema for $g$, denoted by $\text{GR}(g)$, as follows.

$$V_{i+1}(X) = \left(1 - V_i(X)\right) \cdot \min\left\{\frac{1}{2}, g(\cup_{Y \in \text{AH}(X)} \{1 - V_i(Y)\})\right\} + V_i(X) \cdot \max\left\{\frac{1}{2}, g(\cup_{Y \in \text{AH}(X)} \{1 - V_i(Y)\})\right\}$$

If we further assume that $g$ satisfies the optional condition

$(T6)$ If for all $x \in \Delta$, we have that $x < 1$ and for some $x \in \Delta$, $x > 0$, then $g(\Delta) \in (0, 1)$.

Then the above sequence of definitions and theorems in this section still holds if we replace $\text{GR}$ by $\text{GR}(g)$.

The above discussion laid out the properties of the Gabbay-Rodrigues Iteration Schema. In what follows we shall apply it to the following question. Suppose we have an argumentation network $\langle S, R \rangle$ with associated equations and an initial assignment $f : S \mapsto U$. $f$ may come from a single agent who insists on giving certain values to the arguments of $S$; or $f$ may be the result of merging several argumentation frameworks with the nodes in $S$ (through some well-defined process, e.g., voting); or $f$ may arise from any other process. The problem for us is to find the function $f'$, closest to $f$, which is also an extension of $\langle S, R \rangle$ (for example, solves the equations of $\langle S, R \rangle$). Now, what do we mean by “closest”? Following Caminada and Pigozzi [7], we take the view that “closest” means agreeing on the maximal number of nodes with $f$-values in $\{0, 1\}$. In what follows, we show how to find such an assignment $f'$, through the Gabbay-Rodrigues Iteration Schema.

Theorem 2.3 Let $\langle S, R \rangle$ be a network and $f : S \mapsto U$ an assignment of values to the nodes in $S$. Then there is an assignment $h : S \mapsto U$ and unique maximal sets $\text{in}(h) \subseteq \text{in}(f)$ and $\text{out}(h) \subseteq \text{out}(f)$, such that $\text{in}(h)$ is admissible.

Proof. The proof is analogous to the proof of Theorem 5 in [7].

Note that $\emptyset$ is admissible and that $\emptyset \subseteq \text{in}(f)$ and $\emptyset \subseteq \text{out}(f)$. Since $S$ is finite, it only has a finite number of admissible subsets and at least one of them is maximal. First, we show that a maximal admissible subset is also unique. Take any two assignments $g_1$ and $g_2$ such that for all $X \in S$:

- $g_1(X) = 0$ implies $f(X) = 0$ and $g_2(X) = 0$ implies $f(X) = 0$; and
- $g_1(X) = 1$ implies $f(X) = 1$ and $g_2(X) = 1$ implies $f(X) = 1$

It follows that $\text{in}(g_1) \subseteq \text{in}(f)$, $\text{out}(g_1) \subseteq \text{out}(f)$, $\text{in}(g_2) \subseteq \text{in}(f)$ and $\text{out}(g_2) \subseteq \text{out}(f)$. Now suppose that both $\text{in}(g_1)$ and $\text{in}(g_2)$ are maximal and admissible and let us construct an assignment $h : S \mapsto U$, such that

\[
\begin{align*}
h(X) &= 1 \quad \text{iff} \quad \max(g_1(X), g_2(X)) = 1 \\
h(X) &= 0 \quad \text{iff} \quad \min(g_1(X), g_2(X)) = 0 \\
h(X) &= \frac{1}{2} \quad \text{iff} \quad 0 < g_1(X) < 1 \text{ and } 0 < g_2(X) < 1
\end{align*}
\]
Note that the assignment \( h \) is a well-defined function. It is easy to see that every node \( X \) gets at least one value \( h(X) \) and for every node \( X \), this value is unique: \( h(X) \) is equal to \( 1/2 \) if and only if both \( g_1(X) \) and \( g_2(X) \) belong to \( (0,1) \). Now suppose \( h(X) = 1 \). Then \( \max(g_1(X), g_2(X)) = 1 \), then either \( g_1(X) = 1 \) or \( g_2(X) = 1 \), and hence \( f(X) = 1 \). \( h(X) \) cannot be \( 0 \), for if \( \min(g_1(X), g_2(X)) = 0 \), then either \( g_1(X) = 0 \) or \( g_2(X) = 0 \), in which case \( f(X) = 0 \), a contradiction, since \( f \) is a function. Analogously if \( h(X) = 0 \), then \( \min(g_1(X), g_2(X)) = 0 \), and then either \( g_1(X) = 0 \) or \( g_2(X) = 0 \), and hence \( f(X) = 0 \). Therefore, \( h(X) \) cannot also be \( 1 \), for if \( \max(g_1(X), g_2(X)) = 1 \), then either \( g_1(X) = 1 \) or \( g_2(X) = 1 \), in which case \( f(X) = 1 \), a contradiction, since \( f \) is a function.

Note that \( \text{in}(g_1) \subseteq \text{in}(h) \), \( \text{out}(g_1) \subseteq \text{out}(h) \), \( \text{in}(g_2) \subseteq \text{in}(h) \) and \( \text{out}(g_2) \subseteq \text{out}(h) \). We now show that \( \text{in}(h) \) is admissible.

Take any \( X \in \text{in}(h) \), and suppose \( Y \in \text{Att}(X) \). By construction, \( h(X) = 1 \), and hence \( \max(g_1(X), g_2(X)) = 1 \). It follows that i) either \( X \in \text{out}(g_1) \), and since \( \text{in}(g_1) \) is admissible, then \( Y \in \text{out}(g_1) \), and hence \( g_1(Y) = 0 \). \( \min(0, g_2(X)) = 0 \) and hence \( h(Y) = 0 \) and therefore \( Y \in \text{out}(h) \); or ii) \( X \in \text{in}(g_2) \), and since \( \text{in}(g_2) \) is also admissible, \( Y \in \text{out}(g_2) \), and hence \( g_2(Y) = 0 \). \( \min(g_1(Y), 0) = 0 \). Therefore, \( h(Y) = 0 \) and hence \( Y \in \text{out}(h) \).

If on the other hand \( X \in \text{out}(h) \), then \( h(X) = 0 \) and then either \( g_1(X) = 0 \) or \( g_2(X) = 0 \) (or both). That means that either \( X \in \text{out}(g_1) \) or \( X \in \text{out}(g_2) \). Because both of these sets are themselves admissible, we have that there is at least one defeater \( Y \) of \( X \) such that either \( g_1(Y) = 1 \) or \( g_2(Y) = 1 \). It follows that \( h(Y) = \max(g_1(Y), g_2(Y)) = 1 \), and hence \( Y \in \text{in}(h) \). This completes the proof of admissibility.

Notice that \( \text{in}(h) \subseteq \text{in}(f) \) and \( \text{out}(h) \subseteq \text{out}(f) \) as follows. Take \( X \in \text{in}(h) \), then \( h(X) = 1 \), and then \( \max(g_1(X), g_2(X)) = 1 \). Therefore either \( X \in \text{in}(g_1) \) or \( X \in \text{in}(g_2) \) and hence \( X \in \text{in}(f) \). Analogously, if \( X \in \text{out}(h) \), then \( h(X) = 0 \), and then \( \min(g_1(X), g_2(X)) = 0 \). Therefore either \( X \in \text{out}(g_1) \) or \( X \in \text{out}(g_2) \) and hence \( X \in \text{out}(f) \). Since \( \text{in}(g_1) \) and \( \text{out}(g_1) \) are maximal, then \( \text{in}(h) \not\supset \text{in}(g_1) \) and \( \text{out}(h) \not\supset \text{out}(g_1) \). Since \( \text{in}(g_1) \subseteq \text{in}(h) \) and \( \text{out}(g_1) \subseteq \text{out}(h) \), we must have that \( \text{in}(g_1) = \text{in}(h) \) and \( \text{out}(g_1) = \text{out}(h) \). The fact that \( \text{in}(g_2) = \text{in}(h) \) and \( \text{out}(g_2) = \text{out}(h) \) is shown in a similar way. As a result, for any assignment of values \( f : S \rightarrow U \), there is an assignment \( h : S \rightarrow U \) and unique maximal sets \( \text{in}(h) \subseteq \text{in}(f) \) and \( \text{out}(h) \subseteq \text{out}(f) \), such that \( \text{in}(h) \) is admissible.

**Theorem 2.4** Let \( \mathcal{N} = (S, R) \) be a network and \( T \) its GR system of equations. If the sequence of values \( V_0, V_1, \ldots \) becomes stable at iteration \( k \), then \( \text{in}(V_k) \) is the largest admissible set such that \( \text{in}(V_k) \subseteq \text{in}(V_0) \) and \( \text{out}(V_k) \subseteq \text{out}(V_0) \).

**Proof.** We first show that \( \text{in}(V_k) \) is an admissible set.

1. Suppose \( \text{in}(V_k) \) is not conflict-free. Therefore, there must exist \( X, Y \in \text{in}(V_k) \), such that \( (Y, X) \in R \). Since \( X, Y \in \text{in}(V_k) \), \( V_k(X) = V_k(Y) = 1 \). \( V_{k+1}(X) = \max \{ 1/2, 1 - \max_{Y \in \text{Att}(X)} V_k(Y) \} = 1/2 \), and then the sequence is not stable at \( k \), a contradiction. Therefore, \( \text{in}(V_k) \) is conflict-free.
2. Suppose \( \text{in}(V_k) \) is not admissible. It follows that there exists \( X \in \text{in}(V_k) \) and some \( Y \in S \) with \( (Y, X) \in R \), such that \( \text{in}(V_k) \not\rightarrow Y \). Since \( X \in \text{in}(V_k) \), then \( V_k(X) = 1 \) and since the sequence is stable at \( k \), \( V_{k+1}(X) = 1 = \max \{ 1/2, 1 - \max_{W \in \text{Att}(X)} V_k(W) \} \). Therefore, \( \max_{W \in \text{Att}(X)} V_k(W) = 0 \). In particular, \( V_k(Y) = 0 \), and hence \( V_{k+1}(Y) = \min \{ 1/2, 1 - \max_{Z \in \text{Att}(Y)} V_k(Z) \} = 0 \), and therefore there exists \( Z \in \text{Att}(Y) \), such that \( V_k(Z) = 1 \), and hence \( Z \in \text{in}(V_k) \), and hence \( \text{in}(V_k) \rightarrow Y \), a contradiction. Therefore, \( \text{in}(V_k) \) is admissible.

Now we need to show that \( \text{in}(V_k) \) is indeed the maximal admissible set such that \( \text{in}(V_k) \subseteq \text{in}(V_0) \) and \( \text{out}(V_k) \subseteq \text{out}(V_0) \). By Theorem 2.3, there are unique maximal sets \( \text{in}(V_{\max}) \subseteq \text{in}(V_0) \) and \( \text{out}(V_{\max}) \subseteq \text{out}(V_0) \) such that \( \text{in}(V_{\max}) \) is admissible. Furthermore, \( \text{in}(V_{\max}) \supseteq \text{in}(V_k) \) and \( \text{out}(V_{\max}) \supseteq \text{out}(V_k) \).

Suppose either \( \text{in}(V_k) \) or \( \text{out}(V_k) \) are not maximal and let \( 0 < j < k \) be the first index such that there is some \( X \in \text{in}(V_{\max}) \), such that \( X \not\in \text{in}(V_j) \) or that there is some \( Y \in \text{out}(V_{\max}) \) such that \( Y \not\in \text{out}(V_j) \) (or both). We start with the first case. Since \( X \in \text{in}(V_{\max}) \), then \( X \in \text{in}(V_{j-1}) \) and hence \( V_{j-1}(X) = 1 \). Since \( X \not\in \text{in}(V_j) \), then \( V_j(X) < 1 \). It follows that \( V_j(X) = \min \{ 1/2, 1 - \max_{Y \in \text{Att}(X)} V_{j-1}(Y) \} < 1 \). Therefore, there exists \( Y \in \text{Att}(X) \), such that \( V_{j-1}(Y) > 0 \) and hence \( Y \not\in \text{out}(V_{j-1}) \). Since \( \text{in}(V_{\max}) \) is admissible, \( Y \in \text{out}(V_{\max}) \) and this is a contradiction with the fact that \( j \) was the first index such that there was some \( Y \in \text{out}(V_{\max}) \) such that \( Y \not\in \text{out}(V_j) \).

The second case is analogous. Take \( Y \in \text{out}(V_{\max}) \) such that \( Y \not\in \text{out}(V_j) \). Since \( Y \in \text{out}(V_{\max}) \), then \( Y \in \text{out}(V_{j-1}) \) and hence \( V_{j-1}(Y) = 0 \). Since \( Y \not\in \text{out}(V_j) \), then \( V_j(Y) > 0 \). It follows that \( V_j(Y) = \min \{ 1/2, 1 - \max_{Z \in \text{Att}(Y)} V_{j-1}(Z) \} > 0 \). Therefore, for all \( Z \in \text{Att}(Y) \) we have that \( V_{j-1}(Z) < 1 \) and hence there is no \( Z \in \text{Att}(Y) \), such that \( Z \in \text{in}(V_{j-1}) \). Since \( Y \in \text{out}(V_{\max}) \), there must be some \( Z' \in \text{Att}(Y) \), such that \( Z' \in \text{in}(V_{\max}) \), but this is a contradiction since \( Z' \not\in \text{in}(V_{j-1}) \) and \( j \) was the first index such that there was some \( X \in \text{in}(V_{\max}) \), such that \( X \not\in \text{in}(V_j) \).

**Remark 2.2** Given an argumentation network \( N = (S, R) \), an argumentation-friendly function \( g \), a system of equations \( T \) written for \( N \) using \( g \), and an assignment \( v : S \rightarrow U \), which represents initial desired values, then if \( v \) corresponds to a complete extension then the above theorems tell us that the sequence of equations \( V_0 = v \), \( V_1 \), \( V_2 \), ... will become stable at some iteration \( k \) and \( V_k = v \). Otherwise, \( V_k \) is the function giving the maximal possible admissible crisp part in \( (V_k) \) and out \( (V_k) \) agreeing with \( v \). We now have the option of extending \( (V_k) \) into a complete extension \( E_{\text{comp}} \) that is the closest extension agreeing with \( (v) \). This extension would correspond to an assignment \( f' \), which solves the original system of equations \( T \) (by Theorem 1.2).

We will see that if we continue iterating, in the limit of the sequence, we will get an extension.

The following definition helps to translate between values in \( U \) and values in \( \{ \text{in}, \text{out}, \text{und} \} \).


Definition 2.3 (Caminada-Pigozzi/Gabbay-Rodrigues Translation) A labelling function $\lambda$ and a valuation function $V$ can be interdefined according to the table below.

| $\lambda(X)$ → $V_\lambda(X)$ | $V(X)$ → $\lambda_V(X)$ |
|--------------------------------|--------------------------|
| in → 1                         | 1 → in                   |
| out → 0                        | 0 → out                  |
| und → $1/2$                    | $(0, 1)$ → und           |

The choice of the value $1/2$ in the translation from und is arbitrary. Any value in $(0, 1)$ would do, but we will see that legal undecided values will converge to $1/2$ in the limit, and so $1/2$ is the natural choice.

Proposition 2.3 Let $\lambda$ be a labelling function and $V_\lambda$ its corresponding Caminada-Pigozzi translation. If the Gabbay-Rodrigues Iteration Schema is employed using $V_\lambda$ as $V_0$, then for some value $k \geq 0$, the sequence of values $V_0, V_1, \ldots$ will become stable and the sets $\text{in}(V_k)$ and $\text{out}(V_k)$ will correspond to the down-admissible labelling of $\lambda$.

Proof. This follows directly from Theorem 2.4 and Corollary 2.3.

We may also arbitrarily start with $V_0(X) = 1$ for all nodes $X \in S$ and see if this assignment satisfies the equations. At each iteration, the equations may force the crisp values of some nodes to turn to $\text{und}$. Eventually, some iteration $k \leq |S|$ will produce the last set of new undecided values, at which point we say that the sequence has stabilised. We have that $\text{in}(V_k)$ and $\text{out}(V_k)$ correspond to the largest admissible labelling such that $\text{in}(V_k) \subseteq \text{in}(V_0)$ and $\text{out}(V_k) \subseteq \text{out}(V_0)$. $\text{in}(V_k)$ can now form the basis of a complete extension. The smallest of such (complete) extensions comes from what Caminda and Pigozzi called the up-complete labelling of $\lambda$:

Definition 2.4 ([7]) Let $\lambda$ be an admissible labelling. The up-complete labelling of $\lambda$ is a complete labelling $\lambda'$ s.t. $\text{in}(\lambda') \supseteq \text{in}(\lambda)$ and $\text{out}(\lambda') \supseteq \text{out}(\lambda)$ and $\text{in}(\lambda')$ and $\text{out}(\lambda')$ are the smallest sets satisfying these conditions.

If we continue with our calculations we can see what happens with the values $V_0, V_1, \ldots, V_i, \ldots$ in the limit of the sequence. We call these the equilibrium values. Formally,

Definition 2.5 Let $N = (S, R)$ 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 the node $X$ is defined as $V_e(X) = \lim_{i \to \infty} V_i(X)$.

The understanding of the meaning of the equilibrium values requires an analysis of the behaviour of the sequence. The value of a node $X$ is essentially determined by the values of the nodes in $\text{Att}(X)$. At the stable point $k$ we know that the crisp values remain crisp. The values of the attackers of a node at the stable point $k$ can be of one of three types:
1. \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 0 \)
2. \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 1 \)
3. \( 0 < \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} < 1 \)

If the value of a node \( Y \) at the stable point \( k \) is in \( \{0, 1\} \), then Theorem 2.2 ensures that it will remain the same in the limit \( \lim_{i \to \infty} V_i(Y) \). As it turns out, if \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 0 \), then \( \lim_{i \to \infty} V_i(X) = 1 \). And if \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 1 \), then \( \lim_{i \to \infty} V_i(X) = 0 \), as shown by the next theorem.

**Theorem 2.5** Let \( \mathcal{N} = (S, R) \) be an argumentation network and \( V_0 : S \to U \) assign initial values to the nodes in \( S \). Let the sequence of value assignments \( V_0, V_1, V_2, \ldots \) where each \( V_i, i > 0 \), is generated by the Gabbay-Rodrigues Iteration Schema be stable at iteration \( k \). For every \( X \in S \):

1. If \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 0 \), then \( V_e(X) = 1 \); and
2. If \( \max_{Y \in \mathcal{A}(X)} \{ V_k(Y) \} = 1 \), then \( V_e(X) = 0 \).
3. If \( V_k(X) \in \{0, 1\} \), then \( V_e(X) = V_k(X) \);

**Proof.**

1. If \( \max_{Y \in \mathcal{A}(X)} V_k(Y) = 0 \), and the sequence is stable at \( k \), then by Corollary 2.2 \( \max_{Y \in \mathcal{A}(X)} V_{k+1}(Y) = 0 \), for all \( j \geq 0 \). We have that

\[
V_{k+1}(X) = (1 - V_k(X)) \cdot \min\{\frac{1}{2}, 1\} + V_k(X) \cdot \max\{\frac{1}{2}, 1\}
\]

\[
= \frac{1}{2} - \frac{V_k(X)}{2} + V_k(X) = \frac{1}{2} + \frac{V_k(X)}{2}
\]

\[
V_{k+2}(X) = \frac{1}{2} + \frac{1}{4} + \frac{V_k(X)}{4}
\]

\[
V_{k+j}(X) = \sum_{k=1}^{j} \frac{1}{2^k} + \frac{V_k(X)}{2^j}
\]

\[
V_e(X) = \lim_{j \to \infty} V_{k+j}(X)
\]

\[
= \sum_{k=1}^{\infty} \frac{1}{2^k} + \lim_{j \to \infty} \frac{V_k(X)}{2^j} = 1 + 0 = 1
\]

So if the maximum value \( m_k \) of all attackers of \( X \) at iteration \( k \) is 0, then the value of \( X \) converges to 1; and finally
2. If \( \max_{Y \in \text{Att}(X)} V_k(Y) = 1 \), and the sequence is stable at \( k \), then by Corollary 2.2, \( \max_{Y \in \text{Att}(X)} V_{k+j}(Y) = 1 \), for all \( j \geq 0 \). We have that

\[
V_{k+1}(X) = (1 - V_k(X)) \cdot \min\left\{ \frac{1}{2}, 0 \right\} + V_k(X) \cdot \max\left\{ \frac{1}{2}, 0 \right\}
\]

\[
V_{k+2}(X) = \frac{V_k(X)}{4} \quad \Rightarrow \quad V_{k+j}(X) = \frac{V_k(X)}{2^j}
\]

\[
V_e(X) = \lim_{j \to \infty} V_{k+j}(X) = \lim_{j \to \infty} \frac{V_k(X)}{2^j} = 0
\]

So if the maximum value \( m_k \) of all attackers of \( X \) at iteration \( k \) is 1, then the value of \( X \) converges to 0.

3. This follows from the fact that the sequence is stable at \( k \);

The theorem above asserts self-correction for the values of nodes whose attackers are either all out or that have an attacker that is in. Case 3 above, in which \( 0 < \max_{Y \in \text{Att}(X)} \{ V_k(Y) \} < 1 \), is harder and will be dealt with in stages. We start with the case of a cycle whose values of the nodes are all in \((0,1)\) (see Figure 4). Such cycles may involve an even or odd number of nodes, so we have chains of attacks of one of the following types:

- either \( X = Z_1 \leftarrow Z_2 \leftarrow \ldots \leftarrow Z_{2n} = X \) (even cycle)
- or \( X = Z_1 \leftarrow Z_2 \leftarrow \ldots \leftarrow Z_{2n+1} = X \) (odd cycle)

The next lemma shows that in either case, the value of \( X \) in the limit is \( \frac{1}{2} \).

\[
\begin{array}{c}
\vdots \\
Y_1 \\
\vdots \\
Y_i \leftarrow Z_j \leftarrow Z_{j+1} \leftarrow \ldots \leftarrow Z_{2k} \\
Z_1 \leftarrow \ldots \\
Z_2 \leftarrow \ldots \\
Z_3 \leftarrow \ldots \\
Z_4 \leftarrow \ldots \\
\ldots \\
\end{array}
\]

Figure 4:

**Theorem 2.6** Let sequence of values \( V_0, V_1, \ldots \) be stable at iteration \( k \). Consider all possible cycles \( X = Z_1 \leftarrow Z_2 \leftarrow \ldots \leftarrow Z_{2n} = X \) (even) and \( X = Z_1 \leftarrow Z_2 \leftarrow \ldots \leftarrow Z_{2n+1} = X \) (odd) and assume that amongst them we have a cycle such that there exists a sequence of values \( r_1, r_2, \ldots \) such that for each \( Z_i, Z_{i+1} \) is the node in \( \text{Att}(Z_i) \) with maximum value and \( 0 < V_{k+r_1+r_2+\ldots+r_m}(Z_i) < 1 \), for every \( m \geq 0 \). Then \( V_e(Z_i) = \frac{1}{2} \), for all \( Z_i \).

**Proof.** We get the following systems of equations
1. for the cycle $X = Z_1 ← Z_2 ← \ldots ← Z_{2n} = X$:

$V_e(X) = (1 - V_e(X)) \cdot \min\{\frac{1}{2}, 1 - V_e(Y)\} + V_e(X) \cdot \max\{\frac{1}{2}, 1 - V_e(Y)\}$,

where $Y$ is the node in $\text{Att}(X)$ with maximum value. We have two cases to consider.

- $V_e(Y) \geq \frac{1}{2}$, then we get that

  $$V_e(X) = \frac{1 - V_e(Y)}{1.5 - V_e(Y)}$$

- $V_e(Y) \leq \frac{1}{2}$, the we get that

  $$V_e(X) = \frac{1}{1 + 2 \cdot V_e(Y)}$$

it is easy to see from the equations that if $V_e(Y) \geq \frac{1}{2}$, then $V_e(X) \leq \frac{1}{2}$ and if $V_e(Y) \leq \frac{1}{2}$, then $V_e(X) \geq \frac{1}{2}$. Therefore, if we have the cycle $X = Z_1 ← Z_2 ← \ldots ← Z_{2n} = X$, then we get that $\frac{1}{2} \leq Z_1 \leq \frac{1}{2}$, so all $Z_i = \frac{1}{2}$.

2. for the cycle $X = Z_1 ← Z_2 ← \ldots ← Z_{2n+1} = X$, we have that

- either $V_e(Y) \geq \frac{1}{2}$. Let us write $V_e(Y) = \frac{1}{2} + \epsilon(Y)$, for some $0 \leq \epsilon(Y) < \frac{1}{2}$. We then get that

  $$V_e(X) = \frac{1 - V_e(Y)}{1.5 - V_e(Y)}$$

  $$= \frac{1 - \frac{1}{2} - \epsilon(Y)}{1.5 - (\frac{1}{2} + \epsilon(Y))}$$

  $$= \frac{\frac{1}{2} - \epsilon(Y)}{1 - \epsilon(Y)}$$

Write $V_e(X) = \frac{1}{2} - \eta$, for some $0 < \eta < \frac{1}{2}$.

$$\frac{1}{2} - \eta = \frac{\frac{1}{2} - \epsilon(Y)}{1 - \epsilon(Y)}$$

$$\eta = \frac{1 - \frac{1}{2} - \epsilon(Y)}{\frac{1}{2} - \epsilon(Y)}$$

$$= \frac{(1 - \epsilon(Y)) - 2(\frac{1}{2} - \epsilon(Y))}{2(1 - \epsilon(Y))}$$

$$= \frac{1 - \epsilon(Y) - 1 + 2\epsilon(Y)}{2(1 - \epsilon(Y))}$$

$$= \frac{\epsilon(Y)}{2(1 - \epsilon(Y))}$$

24
or $V_e(Y) \leq \frac{1}{2}$. Let us write $V_e(Y) = \frac{1}{2} - \epsilon(Y)$, for some $0 \leq \epsilon(Y) < \frac{1}{2}$.

We then get that

$$V_e(X) = \frac{1}{1 + 2(\frac{1}{2} - \epsilon(Y))} = \frac{1}{1 + 1 - 2\epsilon(Y)} = \frac{1}{2(1 - \epsilon(Y))} = \frac{1}{2} + \eta$$

$$\eta = \frac{1}{2(1 - \epsilon(Y))} - \frac{1}{2} = \frac{1 - 1 + \epsilon(Y)}{2(1 - \epsilon(Y))} = \frac{\epsilon(Y)}{2(1 - \epsilon(Y))}$$

Where are we now? We saw that if we start from $V_e(Y) = \frac{1}{2} \pm \epsilon(Y)$ and $Y \to X$ ($Y$ attacks $X$ as in a cycle), then $V_e(X) = \frac{1}{2} \pm \eta$, where $\eta$ is in the other direction and

$$\eta = \frac{\epsilon(Y)}{2(1 - \epsilon(Y))}.$$ 

Let us now assume a cycle

$$X = Z_1 \leftarrow Z_2 \leftarrow \ldots \leftarrow Z_n = X$$

Assume $Z_1 = \frac{1}{2} \pm \epsilon$. What would the value of $Z_k$ be?

We claim that

$$Z_k = \frac{1}{2} \pm \eta_k$$

where

$$\eta_k = \frac{\epsilon}{2(2^k - (2^k - 1)\epsilon)}$$
The proof is by induction. Let \( X = Z_k \), then \( Y = Z_{k+1} \), and then

\[
\eta_{k+1} = \frac{\eta_k}{2(1 - \eta_k)} = \frac{\eta}{2(2^k - (2^k - 1)e)} = \frac{\eta}{2(2^k - (2^k - 1)e)} = \frac{2(2^k - (2^k - 1)e - \epsilon)}{2(2^k - (2^k - 1)e)} = \frac{2(2^k - (2^k - 1)e - \epsilon)}{2(2^k - (2^k - 1)e)} = \frac{2(2^k - (2^k - 1)e - \epsilon)}{2(2^k + 1 - (2^k - 1)e)} = \epsilon
\]

So the recursion works. Now if we have a loop, we get \( Z_n = Z_1 \)

So \( \eta = \eta_1 \) and thus

\[
\eta = \frac{\eta}{2(2^k + 1 - (2^k - 1)e)}
\]

If we divide by \( \eta (\neq 0) \), we get

\[
1 = \frac{1}{2(2^k + 1 - (2^k - 1)e)}
\]

It is easy to see that only \( \epsilon = \frac{1}{2} \) solves the equation.

**Remark 2.3** Ordinarily we cannot guarantee that \( Z_{i+1} \) is the node in \( \text{Att}(Z_i) \) with maximum value for all \( k' > k \), we need to find a subsequence. This is done as follows: we start with a node \( X \) and since there are a finite number of nodes attacking it (the network is finite), there exists a subsequence such that there is a single attacker whose \( V_{k'} \) value is the maximum for all \( k' \) in the subsequence. We can assume it is \( Z_2 \). This \( Z_2 \) is not unique, there may be other choices. Let \( Z_2^\alpha \) be one arbitrary such choice. Repeating this consideration now for \( Z_2^\alpha \) and for the subsequence thus obtained, we get a \( Z_3^\alpha \) and a further subsequence of the subsequence and so on. Eventually, we get a final subsequence (which depends on the choices of \( Z_i^\alpha \)) \( V_{i+r_1}, V_{i+r_1+r_2}, \ldots \), such that \( Z_{i+1}^\alpha \) is the node in \( \text{Att}(Z_i^\alpha) \) with maximum value and \( 0 < V_{i+r_1+r_2+\ldots+r_m}(Z_{i+1}^\alpha) < 1 \), for each \( m \). It is clear that \( \lim_{i \to \infty} V_i(Z_i^\alpha) = V_{i+r_1} + V_{i+r_2} + \ldots + V_{i+r_m} \text{ for any choice } \alpha_2, \alpha_3, \ldots, \alpha_i \), the subsequence converges to \( 1/2 \), because the \( (\alpha_2, \alpha_3, \ldots) \) represent all possible choices of infinite subsequences of different points \( Z_i^\alpha \) which can participate in the maximum value in \( \text{Att}(Z_i^\alpha) \).

Theorem 2.5 asserts what the limit values of the nodes whose values of the attackers are known at the stable iteration \( k \). Theorem 2.7 asserts the same in terms of the limit values of the attackers.
Theorem 2.7

1. If \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 0 \), then \( V_e(X) = 1 \).

2. If \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 1 \), then \( V_e(X) = 0 \).

Proof. Note that \( \lim_{j \to \infty} \{ V_{j+1}(X) \} = \lim_{j \to \infty} \{ V_j(X) \} \).

1. If \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 0 \), then we have that

\[
V_e(X) = (1 - V_e(X)) \cdot \min\{ \frac{1}{2}, 1 \} + V_e(X) \cdot \max\{ \frac{1}{2}, 1 \}
\]

so if the equilibrium values of all attackers of \( X \) is 0, then the equilibrium value of \( X \) is 1.

2. If \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 1 \), then we have that

\[
V_e(X) = (1 - V_e(X)) \cdot \min\{ \frac{1}{2}, 0 \} + V_e(X) \cdot \max\{ \frac{1}{2}, 0 \}
\]

\[
V_e(X) = \frac{V_e(X)}{2}
\]

\[
e(X) = 0
\]

so if the equilibrium value of any of the attackers of \( X \) is 1, then the equilibrium value of \( X \) is 0.

Theorem 2.8 Let \( \langle S, R \rangle \) be an argumentation network and \( T \) its GR system of equations. If the assignment \( V_0 : S \mapsto U \) is legal then the sequence \( V_0, V_1, V_2, \ldots \), where each \( V_i, i > 0 \), is generated by \( T \), is stable at iteration 0.

Proof. Suppose \( V_0 \) is legal. Then if \( V_0(X) = 0 \), then there exists \( Y \in \text{Att}(X) \) such that \( V_0(Y) = 1 \). Therefore \( V_1(X) = \min\{1/2, 0\} = 0 \). If \( V_0(X) = 1 \), then for all \( Y \in \text{Att}(X) \), \( V_0(Y) = 0 \), and hence \( \max_{Y \in \text{Att}(X)} V_0(Y) = 0 \). Therefore, \( V_1(X) = \max\{1/2, 1\} = 1 \).

The stability of the crisp values then follows from Theorem 2.2, and since \( 0 < V_0(X) < 1 \), then by Theorem 2.7 (case 3), so does the stability of the remaining non-crisp values.

Proposition 2.4 Let \( \langle S, R \rangle \) be an argumentation network; \( T \) its GR system of equations and \( V_e \) a function with the equilibrium values of the nodes in \( S \) calculated according to the Gabbay-Rodrigues Iteration Schema. Let \( \lambda \) be a legal labelling function.
Take any $X \in S$. If $\lambda$ and $V_e$ agree on the values of all nodes in $\text{Att}(X)$, then $\lambda$ and $V_e$ agree on the value of $X$.

**Proof.** There are three cases to consider. Proofs of cases 1. and 2. are similar to the proofs of cases 1. and 2. of Theorem 2.3.

1. $\max_{Y \in \text{Att}(X)} \{V_e(Y)\} = 0$, then for all $Y \in \text{Att}(X)$, $V_e(Y) = 0$. It follows that $V_e(X) = \sum_{k=1}^{\infty} \frac{1}{2^k} + \lim_{j \to \infty} \frac{V_e(X)}{2^j} = 1 + 0 = 1$. Since $V_e$ and $\lambda$ agree with each other on the values of all nodes in $\text{Att}(X)$, we have that for all $Y \in \text{Att}(X)$, $\lambda(Y) = \text{out}$ and since $\lambda$ is legal, $\lambda(X) = \text{in}$, and hence $\lambda$ and $V_e$ agree with each other with respect to the value of $X$ as well.

2. $\max_{Y \in \text{Att}(X)} \{V_e(Y)\} = 1$, then there exists $Y \in \text{Att}(X)$, such that $V_e(Y) = 1$. It follows that $V_e(X) = \lim_{j \to \infty} \frac{V_e(X)}{2^j} = 0$. Since $V_e$ and $\lambda$ agree with each other on the values of all nodes in $\text{Att}(X)$, we have that $\lambda(Y) = \text{in}$ and since $\lambda$ is legal, $\lambda(X) = \text{out}$. Hence $\lambda$ and $V_e$ agree with each other with respect to the value of $X$ as well.

3. $\max_{Y \in \text{Att}(X)} \{V_e(Y)\} = \frac{1}{2}$, then there exists $Y \in \text{Att}(X)$, such that $V_e(Y) = \frac{1}{2}$ (and hence $\lambda(Y) = \text{und}$) and for no $Y \in \text{Att}(X)$, $V_e(Y) = 1$ (and hence for no $Y \in \text{Att}(X)$, $\lambda(Y) = \text{in}$). It follows that

$$V_e(X) = \frac{1 - V_e(X)}{2} + \frac{V_e(X)}{2}$$

$$2 \cdot V_e(X) = 1$$

$$V_e(X) = \frac{1}{2}$$

Since $\lambda$ is legal, $\lambda(X) = \text{und}$, and hence $\lambda$ and $V_e$ agree with each other with respect to the value of $X$.

And now to the main theorem of this section, which explains the the equilibrium values of all nodes and shows their relationship to Caminada and Pigozzi’s down-admissible/up-complete constructions. A down-admissible set is obtained after a series of contraction operations as defined below.

**Definition 2.6 ([7])** Let $\lambda$ be a labelling of an argumentation network $\langle S, R \rangle$. A contraction sequence from $\lambda$ is a sequence of labellings $[\lambda_1 = \lambda, \ldots, \lambda_k]$ such that

1. for each $i \in \{1, \ldots, k - 1\}$, $\lambda_{i+1} = \lambda_i - \{(X, \text{in}), (X, \text{out})\} \cup \{(X, \text{und})\}$, where $X$ is an argument that is illegally labelled $\text{in}$, or illegally labelled $\text{out}$ in $\lambda_j$; and
2. $\lambda_k$ is a labelling without any arguments illegally labelled $\text{in}$ or illegally labelled $\text{out}$.

28
Theorem 6 of \cite{7} shows us that at the end of a contraction sequence $[\lambda_1, \ldots \lambda_k]$, $\lambda_k$ corresponds to the down-admissible labelling of $\lambda_1$, which is the largest admissible labelling that is smaller or equal to $\lambda_1$.

Not even admissible labelling corresponds to a complete extension. An admissible labelling can be turned into a labelling that corresponds to a complete extension by changing the labels of nodes that illegally labelled $\text{und}$, to $\text{in}$ or $\text{out}$ as appropriate. Each such operation is called an expansion, and an expansion sequence corresponds to a list of all such operations:

**Definition 2.7 (\cite{7})** Let $\lambda$ be an admissible labelling of the argumentation network $\langle S, R \rangle$. An expansion sequence from $\lambda$ is a sequence of labellings $[\lambda_1 = \lambda, \ldots \lambda_k]$ such that

1. for each $i \in \{1, \ldots, k - 1\}$,
   \[
   \lambda_{i+1} = \begin{cases} 
   \lambda_i - \{(X, \text{und})\} \cup \{(X, \text{in})\}, & \text{if } X \text{ is an argument that is illegally labelled } \text{und} \text{ in } \lambda_i \text{ and all its attackers are labelled } \text{out} \\
   \lambda_i - \{(X, \text{und})\} \cup \{(X, \text{out})\}, & \text{if } X \text{ is an argument that is illegally labelled } \text{und} \text{ in } \lambda_i \text{ and it has an attacker labelled } \text{in}
   \end{cases}
   \]

2. $\lambda_k$ is a labelling without any arguments illegally labelled $\text{und}$.

Caminada and Pigozzi have shown us that $[\lambda_1, \ldots \lambda_k]$ is an expansion sequence, then $\lambda_k$ is a complete labelling and it is the smallest such labelling containing $\lambda_1$.

We now introduce a few concepts to help us in the proof of the main theorem.

**Definition 2.8** Let $\langle S, R \rangle$ be an argumentation network; $V$ be an assignment of values to the nodes in $S$; and $\lambda$ a labelling of these nodes. We say that $V$ and $\lambda$ agree with each other with respect to the value of a node $X$ if and only if the following conditions hold:

1. $V(X) = 1$ if and only if $\lambda(X) = \text{in}$
2. $V(X) = 0$ if and only if $\lambda(X) = \text{out}$
3. $V(X) = \frac{1}{2}$ if and only if $\lambda(X) = \text{und}$

We say that $V$ and $\lambda$ agree with each other if they agree with the values of all nodes in $S$.

**Definition 2.9 (Attack tree of a node)** Let $\langle S, R \rangle$ be a network. The attack tree of a node $X \in S$ is the tree with root $X$ and for every node $N$ in $\text{Tree}(X)$, the children of $N$ are the nodes in $\text{Att}(N)$.

The attack trees of

**Definition 2.10 (Path from a node)** Let $\langle S, R \rangle$ be a network. Take $X \in S$. A path from $X$ is a sequence of nodes $X = Z_0, Z_1, Z_2, \ldots$ such that each $Z_{i+1}$, $i \geq 0$, is a child of $Z_i$ in the attack tree of $X$. The set of all paths from a node $X$ is denoted $\Pi(X)$. 29
Note that in a SCC $C$ for every path $\pi = Z_0, Z_1, \ldots$ from every node $Z_0 \in C$, there exists a smallest $i(\pi)$ such that for some $r(\pi)$, $Z_{i(\pi)} = Z_{i(\pi) + r(\pi)}$. $i(\pi) < |C|$. $i(\pi)$ is the index of the first node in the path $\pi$ that is involved in a loop, or you can think of it as the minimum distance from the starting node of the path $\pi$ to a looping node in the path. If $i(\pi) = 0$, then $Z_0$ attacks itself. Let us call the loop head of the path $\pi = Z_0, Z_1, \ldots$, the node $Z_{i(\pi)}$.

**Definition 2.11 (V_{\text{max}}\text{-paths})** Let $Z$ be a node in a SCC $C$ and let the sequence of values $V_0, V_1, \ldots$ be stable at iteration $k$. The set of $V_{\text{max}}\text{-paths}$ of $Z$ is defined as $V_{\text{max}}\text{-paths}(Z) = \{ \pi = [Z = Z_0, Z_1, \ldots] \in \Pi(Z) \mid \text{for each } Z_i, V_{k+r}(Z_{i+1}) = \max_{Z_{i+1}} \{ V_{k+r}(Z_{i+1}) \} \text{ for an infinite number of } r \text{'s} \}$. For every $Z \in C$, the set of $V_{\text{max}}\text{-paths}$ from $Z$ is non-empty (see Remark 2.3).

**Definition 2.12 (Bar of a node)** Let $C$ be a SCC and take $X \in C$. The bar of $X$ is the set

$\bar{X} = \{ Z \in C \mid Z \text{ is the loop head of a path in } V_{\text{max}}\text{-paths}(X) \}$. 

**Definition 2.13** Let $\Gamma(X)$ be the set of $V_{\text{max}}\text{-paths}$ of $X$ and take $U \subseteq C$ a set of points. The bar of $X$ modified by $U$ is defined as

$\bar{X}(U) = \bigcup_{\pi \in \Gamma(X)} \{ y \mid y \text{ is the first node in } \pi \text{ such that either } y \text{ is the loop head of } \pi \text{ or } y \in U \}$

**Theorem 2.9** Let $\langle S, R \rangle$ be an argumentation network; $V_0$ be an initial assignment of values to the nodes in $S$; $\lambda_0$ an initial labelling of these nodes; and $V_0$ and $\lambda_0$ faithful to each other according to Definition 2.3. Let $\lambda_{da}$ be the labelling at the end of a contraction sequence from $\lambda_0$ and $\lambda_{CP}$ the labelling at the end of an expansion sequence after $\lambda_{da}$. Let $k$ be the point at which the sequence $V_0, V_1, \ldots$, becomes stable and $V_e(X)$ the equilibrium value of a node calculated through the Gabbay-Rodrigues Iteration Schema. Then $\lambda_{CP}$ and $V_e$ agree with each other according to Definition 2.3.

**Proof.** The proof is done on induction on the depth of a node $X$. Suppose the depth of $X$ is 0. There are three main cases to consider.

Case 1: $X$ is a source node. By definition, $X$ has no attackers, and hence $\max_{Y \in \text{Att}(X)} V_0(Y) = \max_{Y \in \text{Att}(X)} V_k(Y) = 0$ and then by Theorem 2.3 $V_e(X) = 1$.

If $\lambda_0(X) = \text{in}$, then $X$ is legally labelled in. $X$ does not take part in the contraction or expansion sequences and therefore $\lambda_{CP}(X) = \text{in}$. If $\lambda_0(X) = \text{out}$, then $X$ is illegally labelled out, and therefore the label of $X$ is changed to und in the contraction sequence and since it is illegally labelled und, then it is subsequently changed to in in the expansion sequence. If $\lambda_0(X) = \text{und}$, then $X$ cannot be contracted, and since it is illegally labelled und, its label must be changed to in during the expansion sequence. In all cases, $\lambda_{CP}(X) = \text{in}$, and hence $\lambda_{CP}$ and $V_e$ agree with each other with respect to the value of $X$. 


Case 2: \( X \) is part of a source SCC and both \( V_0|C \) and \( \lambda_0|C \) are legal assignments within \( C \). Let us partition \( C \) into two components: \( C^c \) containing all nodes with crisp values and \( C^u \) containing all nodes with undecided values.

Since \( \lambda_0|C \) is a legal assignment, and the nodes in \( C^c \) only have values in \{\text{in, out}\}, then no nodes in \( C^c \) are illegally labelled and hence their labels are unaffected by the contraction sequence. Likewise, since no node is labelled undecided in \( C^c \), nothing can be subsequently expanded and \( \lambda_{CP}|C^c = \lambda_0|C^c \). By construction, the values of all nodes in \( C^u \) are \text{und}, and hence these nodes are not affected by the contraction sequence. Furthermore, they are all legally labelled undecided and hence the values remain unchanged, and hence \( \lambda_{CP}|C = \lambda_0|C \).

Since \( V_0|C \) is a legal assignment, then by Theorem 2.8, it is stable at iteration 0. As a result, for all nodes \( X \subseteq C^c \), \( V_1(X) = V_0(X) \). Hence by Theorem 2.2, \( V_e(X) = V_0(X) \) for all nodes \( X \subseteq C^c \), and then since \( \lambda_0 \) and \( V_0 \) are faithful to each other (Definition 2.3), conditions 1. and 2. of Definition 2.8 are satisfied. We now show that condition 3. also follows.

For all nodes \( X \subseteq C^u \), we have that \( 0 < V_0(X) < 1 \). Since \( V_0|C \) is a legal assignment, then for every \( X \subseteq C^u \), \( 0 < \max_{Y \in Att(X)} \{V_0(Y)\} < 1 \).

Notice that by construction \( C^u = C \setminus C^c \). Stage two of case 3 below shows that for all nodes \( X \subseteq C^u \), \( V_e(X) = 1/2 \). Therefore, condition 3. of Definition 2.8 is also satisfied and as a result, \( \lambda_{CP} \) and \( V_e \) agree with each other with respect to all nodes in \( C \).

Case 3: \( X \) is part of a source SCC and \( \lambda_0|C \) and \( V_0|C \) are not legal assignments.

Stage one:

We know that the sequence of assignments \( V_0, V_1, \ldots \), eventually becomes stable at some iteration \( k \) and by Theorem 2.4, \( \text{in}(V_k) \subseteq \text{in}(V_0) \), \( \text{out}(V_k) \subseteq \text{out}(V_0) \) and \( \text{in}(V_k) \subseteq \text{out}(V_0) \). By Theorem 6 of \([7]\), \( \text{in}(\lambda_{CP}) \) is the largest (and unique) admissible subset of \( \text{in}(\lambda_0) \) and since \( \lambda_0 \) and \( V_0 \) are faithful to each other, we can conclude that \( \text{in}(V_k) = \text{in}(\lambda_{da}) \) and \( \text{out}(V_k) = \text{out}(\lambda_{da}) \).

Note that since the sequence is stable at \( k \), \( \text{in}(V_k) \subseteq \text{in}(V_e) \) and \( \text{out}(V_k) \subseteq \text{out}(V_e) \).

Consider the sequence of expansion operations \( e_1, e_2, \ldots, e_m \) and the sequence of labellings \( \lambda_0 = \lambda_{da}, \lambda_1, \lambda_2, \ldots, \lambda_m = \lambda_{CP} \), where for each \( i > 0 \), \( \lambda_i \) is obtained from \( \lambda_{i-1} \) via the expansion \( e_i \). We show by induction on \( m \) that \( \text{in}(\lambda_{CP}) \subseteq \text{in}(V_e) \) and \( \text{out}(\lambda_{CP}) \subseteq \text{out}(V_e) \). In a second step, we show that if \( \lambda_{CP}(X) = \text{und} \), then \( V_e(X) = 1/2 \).

Suppose that \( e_1 \) turns the node \( X \) illegally labelled \text{und} by \( \lambda_{da} \) into \text{in}.
Then \( \text{out}(\lambda_1) = \text{out}(\lambda_{da}) \) and \( \text{in}(\lambda_1) = \text{in}(\lambda_{da}) \cup \{X\} \). Then for all \( Y \in Att(X) \), \( \lambda_{da}(X) = \text{out} \). Therefore, \( V_k(Y) = 0 \) for all \( Y \in Att(X) \),

\footnote{This effectively means that the only possible incoming attacks from \( C^c \) are from nodes labelled \text{out}. Otherwise, the attacked nodes in \( C^u \) should have been labelled \text{out} and hence would have been illegally labelled \text{und}.}
and hence \( \max_{Y \in \text{Att}(X)} \{ V_k(Y) \} = 0 \). By Theorem 2.5, \( V_e(X) = 1 \) and therefore \( X \in \text{in}(V_e) \). We set \( V_k^{1,\text{out}} = \text{out}(V_k) \) and \( V_k^{1,\text{in}} = \text{in}(V_k) \cup \{ X \} \).

Suppose that \( e_1 \) turns the node \( X \) illegally labelled \text{und} by \( \lambda_{da} \) into \text{out}. Then \( \text{in}(\lambda_1) = \text{in}(\lambda_{da}) \) and \( \text{out}(\lambda_1) = \text{out}(\lambda_{da}) \cup \{ X \} \). Then there exists \( Y \in \text{Att}(X) \) such that \( \lambda_{da}(X) = \text{in} \). Therefore, \( V_k(Y) = 1 \) for some \( Y \in \text{Att}(X) \), and hence \( \max_{Y \in \text{Att}(X)} \{ V_k(Y) \} = 1 \). By Theorem 2.5, \( V_e(X) = 0 \) and therefore \( X \in \text{out}(V_e(X)) \). We set \( V_k^{1,\text{out}} = \text{out}(V_k) \cup \{ X \} \) and \( V_k^{1,\text{in}} = \text{in}(V_k) \).

Assume that for some \( i \), \( \text{in}(\lambda_i) = V_k^{i,\text{in}} \) and \( \text{out}(\lambda_i) = V_k^{i,\text{out}} \). Now consider the \( i+1 \)-th expansion operation \( e_{i+1} \).

Suppose that \( e_{i+1} \) turns the node \( X \) illegally labelled \text{und} in \( \lambda_i \) into \text{in}. Then for all \( Y \in \text{Att}(X) \), \( \lambda_i(X) = \text{out} \). Therefore, \( V_e(Y) = 0 \) for all \( Y \in \text{Att}(X) \), and hence \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 0 \). By Theorem 2.5, \( V_e(X) = 1 \) and therefore \( X \in \text{in}(V_e) \). As before, we set \( V_k^{i+1,\text{out}} = V_k^{i,\text{out}} \) and \( V_k^{i+1,\text{in}} = \text{in}(V_k) \).

Suppose that \( e_{i+1} \) turns the node \( X \) illegally labelled \text{und} by \( \lambda_i \) into \text{out}. Then there exists \( Y \in \text{Att}(X) \) such that \( \lambda_i(X) = \text{in} \). Therefore, \( V_e(Y) = 1 \) for some \( Y \in \text{Att}(X) \), and hence \( \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} = 1 \). By Theorem 2.5, \( V_e(X) = 0 \) and therefore \( X \in \text{out}(V_e(X)) \). Again, we set \( V_k^{i+1,\text{out}} = V_k \cup \{ X \} \) and \( V_k^{i+1,\text{in}} = V_k^{i,\text{in}} \).

By now we know that if \( X \in V_k^{m,\text{in}} \), then \( V_e(X) = 1 \) and \( \lambda_{CP}(X) = \text{in} \) and that \( X \in V_k^{m,\text{out}} \), then \( V_e(X) = 0 \) and \( \lambda_{CP}(X) = \text{out} \). We ask if there is some \( Z \notin V_k^{m,\text{in}} \) such that \( V_e(Z) = 1 \) or \( Z \notin V_k^{m,\text{out}} \) such that \( V_e(Z) = 0 \). The answer is no as it is explained in stage two below.

Stage two:
Let us use \( C^c \) to denote \( \{ V_k^{m,\text{in}} \cup V_k^{m,\text{out}} \} \) and \( C^u \) to denote \( C \setminus C^c \). Suppose \( X \in C^u \).

We know that \( V_k^{m,\text{in}} = \text{in}(\lambda_{CP}) \) is a complete extension and that no further expansion operation is possible from \( \lambda_{CP} \), therefore if \( X \notin \text{in}(\lambda_{CP}) \), then either \( \lambda_{CP}(X) = \text{out} \) and hence \( X \in V_k^{m,\text{out}} \), which is not possible, or \( \lambda_{CP}(X) = \text{und} \) and legally so. Therefore there exists \( Y \in \text{Att}(X) \), such that \( \lambda_{CP}(Y) = \text{und} \) and hence \( 0 < \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} < 1 \).

Similarly, if \( X \notin \text{out}(\lambda_{CP}) \), then either \( \lambda_{CP}(X) = \text{in} \) and hence \( X \in V_k^{m,\text{in}} \), which is not possible, or \( \lambda_{CP}(X) = \text{und} \) and legally so. Therefore there exists \( Y \in \text{Att}(X) \), such that \( \lambda_{CP}(Y) = \text{und} \) and hence \( 0 < \max_{Y \in \text{Att}(X)} \{ V_e(Y) \} < 1 \) and therefore \( 0 < V_e(X) < 1 \).

So we know that for all \( X \in C^u \), \( \lambda_{CP}(X) = \text{und} \) and \( 0 < V_e(X) < 1 \). In what follows, we will show that indeed for all nodes in \( C - C^c \), \( V_e(X) = \frac{1}{2} \). Note that since we are in a SCC \( C \), for all \( X \in C^u \), there is an infinite attack tree with root \( X \), in which every branch is of the form \( X = Z_0, Z_1, Z_2, \ldots, Z_k = X \), where for every \( i > 0 \), \((Z_{i+1}, Z_i) \in R \). Some
of the $Z_i$ are in $V_k^{m,\text{out}}$, but none can be in $V_k^{m,\text{in}}$, for that would make $Z_{i-1}$ out.

The proof is done by induction on the maximum distance from a node $X$ in $C^n$ to a loop $Z_1, Z_2, \ldots, Z_k = Z_1$, where every $Z_i \in C\setminus V_k^m$. There are infinitely many paths from $X$ in the attack tree of $X$, but we only need to consider the set $\Gamma(X)$ with all $V_{\max}$-paths of $X$. Each such path is of the form $\pi(X) = (Z_0 = X), Z_1, \ldots$. Now define the distance of $X$, $\dim X$, as the maximum index $i$ such that for each path $\pi(X)$, $Z_i \in \text{bar}(Z, V_k^{m,\text{out}})$. This means that $Z_i$ is the first point in the path $\pi(X)$ which is either a repetition of a previous point or a point in $V_k^{m,\text{out}}$.

If $\dim X = 0$, then $X$ must be attacked by a cycle involving only $X$ (otherwise $X \in V_k^{m,\text{out}}$, and then $V_e(X) = 0$, a contradiction). Therefore, we have a cycle that attacks $X$ and which involves $X$ alone. All attackers in this cycle (i.e., $X$) have maximum value and $0 < V_{k+r}(X) < 1$ for every $r \geq 0$. By Theorem 2.6, the value of every node in the cycle is $V_e(X) = \frac{1}{2}$. Now the equilibrium value of the node $X$ attacked by the cycle is calculated by

$$V_e(X) = (1 - V_e(X)) \cdot \min\{\frac{1}{2}, \frac{1}{2}\} + V_e(X) \cdot \max\{\frac{1}{2}, \frac{1}{2}\}$$

$$= \frac{1 - V_e(X)}{2} + \frac{V_e(X)}{2}$$

$$= \frac{1 - V_e(X) + V_e(X)}{2}$$

$$= \frac{1}{2}$$

Now assume that the equilibrium value of all nodes with distance up to $k$ is $\frac{1}{2}$ and consider the node $X$ with distance $k+1$. For all $Y \in \text{Att}(X)$, we have that $\dim Y \leq k$. Therefore, either $Y \in V_k^{m,\text{out}}$ in which case $V_e(Y) = 0$, or by the inductive hypothesis $V_e(Y) = \frac{1}{2}$

Therefore we have that $\max Y \in \text{Att}(X)\{V_e(Y)\} = \frac{1}{2}$ and as before

$$V_e(X) = (1 - V_e(X)) \cdot \min\{\frac{1}{2}, \frac{1}{2}\} + V_e(X) \cdot \max\{\frac{1}{2}, \frac{1}{2}\}$$

$$= \frac{1}{2}$$

To conclude, for all $X \in V_k^{m,\text{in}}, V_e(X) = 0$; for all $X \in V_k^{m,\text{out}}, V_e(X) = 0$; and for all $X \in C^n, V_e(X) = \frac{1}{2}$. in($V_e[C]$) (resp., in(\lambda_{CP}[C]) in this case is the minimal complete extension containing in($V_k[C]$) (resp., in(\lambda_{da}[C])).

Assume the theorem holds for all nodes of depth up to $k$. We now show that it holds for nodes of depth $k+1$.

Note that $\text{Att}(X) \not\subseteq V_k^{m,\text{out}}$, otherwise $X$ would be illegally labelled und.
Define $\text{Known}_k^0 = \{X \in S \mid \text{depth}(X) \leq k\}$ and $\text{Known}_k^{m+1} = \{X \in S \mid \text{depth}(X) = k+1\}$ and for all $Y \in \text{Att}(X), Y \in \text{Known}_k^n\}$.

We show that for all $i \geq 0$, we have that $\lambda_{CP}(X) = V_e(X)$, for all $X \in \text{Known}_k^i$. First notice that by induction hypothesis, $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_k^{i-1}$. Now suppose that $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_k^i$, then by Proposition 2.4 $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_k^{i+1}$. Since the network is finite, $\text{Known}_k^i = \text{Known}_k^{e+1}$, for some $e \geq 0$. Define $C_{k+1}^u = \{X \in S \mid \text{depth}(X) = k+1\} \setminus \text{Known}_k^{e+1}$.

By definition, if there exists $X \in C_{k+1}^u$ and $Y \in \text{Att}(X)$ such that $Y \in \text{Known}_k^{e+1}$, then $\lambda_{CP}(Y) = \text{out}$ and $V_e(Y) = 0$ (otherwise the value of $X$ would be known). Therefore, we can exclude the nodes in $\text{Known}_k^{e+1}$ and consider $C_{k+1}^u$ in isolation. $C_{k+1}^u$ can therefore be treated as a network of depth 0, and the proof will follow exactly from Cases 2 and 3 of the base of the main induction, and hence for all $X \in C_{k+1}^u$, $V_e(X) = \lambda_{CP}(X)$.

**Corollary 2.4** Let $\langle S, R \rangle$ be an argumentation network and $V_0$ be an initial assignment of values to the nodes in $S$. Let $V_e(X)$ be the equilibrium value of a node $X$ calculated through the Gabbay-Rodrigues Iteration Schema. For all nodes $X \in S$, $V_e(X) \in \{0, 1/2, 1\}$.

**Proof.** Follows from the possible equilibrium values of all nodes in Theorem 2.7.

### 3 Discussion and Worked Examples

Suppose we are given a network such as the one in Figure 5 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, we consider three sets of representative initial values 1, 2, and 3. The table in Figure 5 shows what happens when these values are applied to the equations, giving both the values at the stable point ($V_k$) and at the limit ($V_e$). The corresponding down-admissible labellings and

|   | $X$ | $Y$ | $W$ | $Z$ |
|---|-----|-----|-----|-----|
| 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, 3/8, 1/2) |
| 3 | (1, 1, 1) | (0, 0, 0) | (1, 1, 1) | (0, 0, 0) |

![Network used in Section 3](image-url)
their resulting up-completion according to Caminada-Pigozzi’s procedure can be obtained simply by replacing 0 with \textbf{out}, 1 with \textbf{in} and values in (0, 1) with \textbf{und}.

Case 1. represents the situation in which the initial values in the cycle $W \leftrightarrow Z$ are compatible with an extension and hence the crisp values are preserved by the calculations. We end up with the complete extension $E_1 = \{X, Z\}$. Constrast this with case 2., in which the initial values of $W$ and $Z$ are 1 and $Z$, respectively. 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 $E_2 = \{X\}$. This interference does not happen in case 1., because there the undecided value of $Y$ is dominated by $Z$’s stronger value. Since we use max, $W$’s initial value of 0 is retained and hence so is $Z$’s.

If however we start with a preferred extension, which is also complete by definition, we get as a result unchanged initial values (cf. Theorem 2.9). Caminada-Pigozzi also give the same result because the down-admissible 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 5).

We can suggest an enhanced procedure to improve on the results obtained in case 2., which is outlined below. The procedure starts with an empty set of crisp values ($\textbf{Crisp}$) and a set of initial values to the nodes.

1. Calculate the equilibrium values for all nodes using the iteration schema.
2. If $\{X \in S \mid V_e(X) \in \{0, 1\}\} \subseteq \textbf{Crisp}$, stop. The extension is defined in the set $\{X \mid V_e(X) = 1\}$. Otherwise, set $\textbf{Crisp} = \textbf{Crisp} \cup \{X \in S \mid V_e(X) \in \{0, 1\}\}$ and proceed to step 3.
3. For every $X \in \{X \mid V_e(X) \in \{0, 1\}\}$, 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 \textit{sound}, since at each run the equilibrium values computed yield a complete extension. Note that re-using some of the original values does not affect soundness. If they cannot be used to generate a larger extension, they will just converge to $1/2$. The procedure also \textit{terminates} as long as the original network $S$ is finite, since a new iteration 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, $\textbf{Crisp} = \{X, Y\}$. We then run it once more, this time with initial values $V_0(X) = 1$, $V_0(Y) = 0$, $V_0(W) = 1$ and $V_0(Z) = 0$. This will stabilise immediately at these values and then $\textbf{Crisp} = \{X, Y, W, Z\}$. In the third run, no new crisp values are generated,
so we stop with extension \( \{X, W\} \), which is a preferred extension (see case 3. above). This is closer to the original values, because the preference of \( W \) over \( Z \) is preserved.

3.1 Worked Examples with Cycles

The table in Figure 6 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 iteration in which the stable and equilibrium values were reached (S,E). Obviously the equilibrium values are an approximation. 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. Independent nodes, such as \( Z \) in the networks above always converge to 1 independently of their initial values. This also happens to all nodes whose values of the attackers all converge to 0. Cases (L) and (R) explore different scenarios involving cycles. The odd cycle in (L) attacks the even cycle \( X \leftrightarrow Y \) and the even cycle in (R) attacks the odd cycle \( A \rightarrow B \rightarrow C \rightarrow A \).

We start with (L), which contains an odd cycle attacking an even cycle. The values in the odd cycle in this case will converge to \(1/2\) independently of their initial values. This may or may not have an effect on nodes that are attacked by any of the nodes in the cycle. We start with an initial valid configuration for \( X \) and \( Y \) in both (L1) and (L2). The end results will differ though as explained next. If \( X \) starts with 0 and \( Y \) with 1 (L1), then the undecidedness of \( B \) is dominated by the value of \( Y \) and the original values persist. However, if \( X \) starts with 1 and \( Y \) with 0, the undecidedness of \( B \) forces \( X \) to become undecided, which in turns makes \( Y \) also become undecided. As a result, all of the values converge to \(1/2\) apart from \( Z \), which as we said is independent and converges to 1 (L2).

Now let us look at (R) in which the even cycle attacks the odd one. (R1) and (R2) contain different initial valid configurations for the even cycle. This time the nodes in the even cycle are independent of external values and their original values remain. If \( X \) starts with 1, it remains with 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 its values will converge to \(1/2\) independently of their initial values (R2).

4 Fine analysis of the Gabbay-Rodrigues formula

This section studies the detailed properties of the Gabbay-Rodrigues formula in terms of traditional extensions in Dung’s sense. We need to develop a sequence

\[\text{Effectively this means that if the maximum variation in node values between two successive iterations is smaller than } 10^{-19}, \text{ we cannot be sure it is not simply the result of a rounding error due to the precision of the computer. At that point we assume we have reached the limit of what can be accurately calculated.}\]
of definitions and concepts and do our comparisons relative to these concepts.

Definition 4.1 Let \( \langle S, R \rangle \) be a network. Let \( \langle S^*, R^* \rangle \) be the acyclic network of all strongly connected components (SCCs) of \( \langle S, R \rangle \). The means the following for \( s^* \in S^*:

1. \( s^* \subseteq S \) and \( \bigcup_{s \in S^*} s = S \) (considering \( s^* \in S \) and the union of the nodes in \( s^* \))

2. Each \( s^* \) is either a single point \( X \) such that \( (X, X) \notin R \), or a maximal complete loop, i.e., a set \( s^* \) which satisfies that for all \( X, Y \in s^* \) there exist \( X_1, X_2, \ldots, X_n \) such that \( (X, X_i) \in R \) and \( (X_n, Y) \in R \) and \( s^* \) is maximal with respect to this property.

3. If \( (s^*, t^*) \in R^* \) then both \( 3a \) and \( 3b \) hold:
   
   (a) for some \( X^* \in s^* \), \( Y \in t^* \), \( (X, Y) \in R \)
   
   (b) for no \( U, V \) do we have \( U \in s^* \), \( V \in T^* \) and \( (V, U) \in R \).

4. Note that \( s^* \neq t^* \) implies \( s^* \cap t^* = \emptyset \), because of the maximality condition.

Example 4.1 Let \( \langle S_0, R_0 \rangle \) be as given in Figure 7. The corresponding acyclic network of all strongly connected components is the one given in Figure 7.

Definition 4.2 Let \( \langle S, R \rangle \) be a network.
1. The function $g : S \rightarrow \{0, 1\}$ is a partial extension function if the domain of $g$, $\text{dom} g \subseteq S$; the range of $g$ is $\{0, 1\}$; and $g$ satisfies the following conditions:
   
   i) if for some $Y \in \text{dom} g$ we have $g(Y) = 1$ and $(Y, X) \in R$, then $X \in \text{dom} g$ and $g(X) = 0$
   
   ii) if for all $Y$ such $(Y, X) \in R$ we have that $Y \in \text{dom} g$ and $g(Y) = 0$, then $X \in \text{dom} g$ and $g(X) = 1$

2. A partial extension function $g$ is said to be generated by $\gamma_0(g) \subseteq \text{dom} g$, if and only if the following holds:
   
   i) $\gamma_0(g)$ is a conflict-free set of points in $\langle S, R \rangle$
   
   ii) If $Y \in \gamma_0(g)$, then $g(Y) = 1$
iii) \( \text{dom } g \) is the closure \( \gamma(g) \) of \( \gamma_0(g) \) under the following operations, i.e., the smallest set containing \( \gamma_0(g) \) and closed under the operations a)–c) below:

a) \( \gamma_0(g) \subseteq \gamma(g) \).

b) If \( X \in S \) and for some \( Y \in \gamma(g) \), \( g(Y) = 1 \) and \( (Y, X) \in R \), then \( X \in \gamma(g) \) and \( g(X) = 0 \).

c) If \( X \in S \) and for all \( Y \) such that \( (Y, X) \in R \), we have that \( Y \in \gamma(g) \) and that \( g(Y) = 0 \), then \( X \in \gamma(g) \) and \( g(X) = 0 \).

**Definition 4.3** Let \( (S_0, R_0) \) be a network. Construct \( (S_0^\ast, R_0^\ast) \). We use \( (S_0^\ast, R_0^\ast) \) to define a family of extension functions. These will give us all complete extensions of \( (S_0, R_0) \). The functions will be defined by the inductive operations below.

**Inductive operation:**

Let \( (s_1^\ast, \ldots, s_k^\ast) \) be the source nodes of \( (S_0^\ast, R_0^\ast) \). Let \( \alpha_0 = (E_1^{0,0}, \ldots, E_k^{0,0}) \) be a choice of complete extensions of \( (s_i^\ast, R|s_i^\ast) \). There may be several such choices. Then the following can be the case, for any choice \( \sigma^0 \):

1. \( s_i^\ast = \{X\} \) and \( (X, X) \notin R \), then \( E_{i,0}^\ast = \{X\} \).

2. \( s_i^\ast \) is a loop which has no non-empty complete extension, then \( E_{i,0}^\ast = \emptyset \) (being the all-undecided grounded extension of the network \( (s_i^\ast, R|s_i^\ast) \)).

3. \( s_i^\ast \) is an even loop which does have non-empty extensions and \( E_{i,0}^\ast \) is one of them or \( E_{i,0}^\ast = \emptyset \)

The above basically says that \( \sigma^0 \) is a choice of complete extensions \( E_{i,0}^\ast \) for the network \( (s_i^\ast, R|s_i^\ast) \), for \( i = 1, \ldots, k \), respectively.

For each such \( E_{i,0}^\ast \), let \( E_i^{0,+,0} = \{X | X \in \text{in}\} \). Let \( \sigma^{0,+,0} = \bigcup_i E_i^{0,+,0} \). Let \( g_0^0(\alpha) \) be the function generated by \( \sigma^{0,+,0} \). The superscript 0 says that we are dealing with the network \( (S_0, R_0) \).

Note that \( g_0^0(\alpha) \) might be \( \emptyset \) if all source nodes of \( (S_0^\ast, R_0^\ast) \) are odd loops.

**Lemma 4.1** \( g_0^0(\alpha) \) is a partial extension function (may be empty).

**Proof.** Straightforward.

We continue with Definition 4.3

**Definition 4.3 (continued)** Call all functions \( g_0(\sigma) \), for such a choice \( \sigma \) level 0 functions. Let \( \sigma_0 \) be fixed. Let \( S_1 = S_0 - \text{dom } g_0^0(\sigma) \) and \( R_1 = \{(X, Y) \in R_0 | X, Y \in S_1\} \). If \( S_1 = S_0 \) (i.e., \( g_0^0(\sigma) = \emptyset \)), then stop. Otherwise construct \( (S_1^\ast, R_1^\ast) \). Repeat the inductive operation construction above for \( (S_1^\ast, R_1^\ast) \) for a choice \( \sigma_1 \) of extensions. Construct all functions \( g_1^1(\sigma_1) \) (again, the superscript 1 indicates this is for \( (S_1^\ast, R_1^\ast) \)). Define \( g_{0,1}(\sigma^0, \sigma^1) \) to be the union of \( g_0^0(\sigma_0) \) and \( g_1^1(\sigma_1) \).

**Inductive step:**
Figure 9: \((S_1, R_1)\) for the network in Figure 7

Suppose we defined \((S_0, R_0)\), \ldots, \((S_k, R_k)\) and have a corresponding sequence \(\sigma_0, \sigma_1, \ldots, \sigma_k\) and \(g^0(\sigma_0), g^1(\sigma_1), \ldots, g^k(\sigma_k)\) and \(g_{0,1,\ldots,i}(\sigma_0, \sigma_1, \ldots, \sigma_i), 1 \leq i \leq k\), are all defined and \((S_{k+1}, R_{k+1})\) is also defined. The repeat the construction for \((S_{k+1}, R_{k+1})\) and get \(g^{k+1}(\sigma_{k+1})\) and \(g_{0,1,\ldots,i}(\sigma_0, \sigma_1, \ldots, \sigma_{k+1})\), etc.

**Final step:**

At some point \(m\), we will either stop because \(\text{dom } g^m = \emptyset\) or we will stop because \(S_{m+1} = \emptyset\).

**Example 4.2** Let us examine Figures 7 and 8. The source loops of \((S_0, R_0)\) are \(s_1 = \{A, B, C\}\) and \(s_2 = \{E\}\). There is only one \(\sigma_0\). \(E_1^{0,0} = \emptyset\) and \(E_1^{1,0} = \{E\}\).

The function \(g^0(\sigma_0)\) is defined on \(\{E, W, Z, U\}\) with \(g^0(E) = g^0(Z) = 1\) and \(g^0(W) = g^0(U) = 0\). \((S_1, R_1)\) is therefore the one shown in Figure 7.

The top nodes are \(t_1^* = \{A, B, C\}, t_2^* = \{X, Y\}\) and \(t_3^* = \{F, G\}\). There are 9 possible vectors of extensions:

| \sigma | \(t_1^*\) | \(t_2^*\) | \(t_3^*\) |
|--------|----------|----------|----------|
| \sigma_1 | \emptyset | \{X\} | \{F\} |
| \sigma_2 | \emptyset | \{X\} | \{G\} |
| \sigma_3 | \emptyset | \{X\} | \emptyset |
| \sigma_4 | \emptyset | \{Y\} | \{F\} |
| \sigma_5 | \emptyset | \{Y\} | \{G\} |
| \sigma_6 | \emptyset | \{Y\} | \emptyset |
| \sigma_7 | \emptyset | \emptyset | \{F\} |
| \sigma_8 | \emptyset | \emptyset | \{G\} |
| \sigma_9 | \emptyset | \emptyset | \emptyset |

We have 9 functions \(g^i(\sigma_1^i), i = 1, \ldots, 9\) and nine extensions \(g_{0,1}(\sigma_0, \sigma_1^i), i = 1, \ldots, 9\).
Lemma 4.2 Let $\langle S, R \rangle$ be a network and let $E$ be an extension. Then there exist $\sigma^0, \sigma^1, \ldots, \sigma^k$ such that $E$ is $g_{0,\ldots,k}(\sigma^0, \ldots, \sigma^k)$.

Proof. We follow the above construction process and whenever we need to choose extensions for the source cycles we use $E$.

Definition 4.4 Let $\langle S_0, R_0 \rangle$ be a network and let $E$ be a complete extension. Then for some $k$ and $\sigma^0, \sigma^1, \ldots, \sigma^k$ we have that $E$ is characterised by $g_{0,\ldots,k}(\sigma^0, \ldots, \sigma^k)$. Let $X, Y \in S_0$ such that $\langle X, Y \rangle \in R_0$. We say that $(X, Y)$ is a critical pair at level $0 \leq m \leq k$, if the following hold:

1. $X \in S_m$
2. $S \notin S_{m+1}$
3. $Y \in S_{m+1}$
4. $g_{0,\ldots,m+1}(\sigma^0, \ldots, \sigma^{m+1})(Y) = 1$

Remark 4.1 We shall see that the Gabbay-Rodrigues Iteration Schema can give you extensions but not the one which have critical points. If we examine Figures 7, 8 and 9 as discussed in Example 4.2, then the extensions $\{E, W, Z, U, F\}$ and $\{E, W, Z, U, F, X\}$ have $(U, F)$ as a critical pair. Also $\{E, W, Z, U, Y, G\}$ have $(U, Y)$ as a critical pair and $\{E, W, Z, U, Y, F\}$ have the two critical pairs $(U, Y)$ and $(U, F)$.

We now explain what extensions does the Gabbay-Rodrigues iteration formula give.

Definition 4.5 Let $\langle S_0, R_0 \rangle$ be a network and $\lambda : S \rightarrow \{0, \frac{1}{2}, 1\}$ be an initial function representing a set of desired values for $S$. $\lambda$ may or may not correspond to a legitimate extension.

We define a complete extension using $\lambda$ which we shall call the Gabbay-Rodrigues $\lambda$-accommodating complete extension (denoted by GR($\lambda$)) as follows.

Consider $\langle S_0, R_0 \rangle$ defined for $\langle S_0, R_0 \rangle$ according to Definition 4.2. We shall define $\sigma_0(\lambda)$ as in Definition 4.3 using $\lambda$.

Consider the source components $s^*_1, \ldots, s^*_k$ as discussed in Definition 4.3. There are the following possibilities for any $s^*_i$, as outlined in Definition 4.3.

1. $s^*_i = \{X\}$, for some $X$, such that $(X, X) \notin R_0$. We let $E_{i,\lambda} = \{X\}$.
2. $s^*_i$ is a cycle. If $\lambda \upharpoonright s^*_i$ is a proper extension $E^*_{i,0}$ of $\langle s^*_i, R_0 \upharpoonright s^*_i \rangle$, then let $E^*_{i,\lambda}$ be this extension.
3. If $\lambda \upharpoonright s^*_i$ is not a legitimate extension, then let $E^*_{i,\lambda} = \emptyset$.

We now have our $\sigma^0(\lambda)$. We can proceed as in Definition 4.2 and consider $\langle S_1^*, R^*_1, 1 \rangle$.

We consider $\langle S_1, R_1 \rangle$. We look at $\langle S_1^*, R^*_1, 1 \rangle$ and consider the source cycles $t^*_1, \ldots, t^*_m$.

Claim: There are no top cycles of the form $\{X\}$ with $(X, X) \notin R_0$. Given a proper source cycle $t^*_i$, there are several options.
a) \( \lambda \upharpoonright t^*_i \) is not a proper complete extension of \((t^*_i, R \upharpoonright t^*_i)\). In this case, let \( E^{i, 1}_{t^*_i} = \emptyset \).

b) \( \lambda \upharpoonright t^*_i \) is a proper complete extension, but for some \( Y \in t^*_i \), \( \lambda(Y) = 1 \) and for some \( X \in S - S_1 \), \( g^0(\sigma^0(\lambda))(X) = 0 \), i.e., \((X, Y)\) is a critical pair. In this case again, we let \( E^{i, 1}_{t^*_i} = \emptyset \).

c) \( \lambda \upharpoonright t^*_i \) is a proper extension and there does not exist critical pairs as in \((4.5)\) above. In this case, let \( E^{i, 1}_{t^*_i} = \lambda \upharpoonright t^*_i \).

Let \( \sigma^1(\lambda) \) be defined using \( E^{i, 1}_{t^*_i} \). If \( \sigma^1(\lambda) = \emptyset \), then we stop. Otherwise we continue the process until for some \( m \), we have either \( \sigma(\lambda) = \emptyset \) or \( S_{m+1} = \emptyset \). We now define \( GR(\lambda) \) using \( g_{1, ..., m}(\sigma^0(\lambda), ..., \alpha^m) \) as follows. Let \( GR(\lambda) \) be the extension defined by the function \( \rho : S \rightarrow \{0, \frac{1}{2}, 1\} \) where

\[
\rho(X) = \begin{cases} 
  g_{1, ..., m}(\sigma^0(\lambda), ..., \alpha^m), & \text{if } X \in \text{dom } g_{1, ..., m}(\sigma^0(\lambda), ..., \alpha^m) \\
  \frac{1}{2}, & \text{otherwise}
\end{cases}
\]

Theorem 4.1 Let \( \langle S_0, R_0 \rangle \) be a network and \( \lambda \) be an initial value function. Apply the Gabbay-Rodrigues Iteration Schema, then the values in \( \lim_{j \to \infty} V_j(X) \) correspond to a complete extension \( GR(\lambda) \) of Definition 4.5.

Example 4.3 Let us see what happens if we take initial values for \( \langle S_0, R_0 \rangle \) of Figure 7. Let \( \lambda \) be as follows.

\[
\lambda(A) = \lambda(B) = \lambda(C) = \lambda(X) = \lambda(F) = 1 \\
\lambda(E) = \lambda(W) = \lambda(Z) = \lambda(Y) = \lambda(G) = 0
\]

We apply the procedure of definition 4.3, we get the complete extension \( \{E, Z, X\} \). This is not the best possible extension close to \( \lambda \) because we could have added \( F \) to it to get \( \{E, Z, X, F\} \). However our procedure finds the pair \((U, F)\) critical and so we have to make \( F = G = \text{undecided} \).

The next remark improves on the \( GR(\lambda) \) procedure.
We start with \( \lambda \) and \( \langle S_0, R_0 \rangle \). We compute \( GR(\lambda) \). We now look at \( S_{\lambda} = S_0 - \{X \upharpoonright GR(\lambda)(X) \in \{0, 1\}\} \).

We look at \( \lambda_1 = \langle S_{\lambda}, R \upharpoonright S_{\lambda} \rangle \) and consider \( \lambda \upharpoonright S_{\lambda} \). We repeat the \( GR \)-procedure to \( \lambda_1 \) and \( \langle S_{\lambda}, R \upharpoontright S_{\lambda} \rangle \). We keep iterating this procedure until there is no change. If we execute this procedure for Figure 7 we get a new system in Figure 10. The procedure for this system will accept \( F = 1, G = 0 \).

Definition 4.6 How to use the \( GR \)-procedure to find the best possible complete extension that matches a given set of initial values \( \lambda \).

Given \( A_0 = \langle S_0, R_0 \rangle \) and \( \lambda \), we want to define \( GRB(\lambda) \).

step 1): Define \( GR(\lambda, A_0) \).

step 2): Let \( A_1 = A_0 - GR(\lambda, A_0) \), where \( A_1 = \langle S_{\lambda}, R \upharpoonright S_{\lambda} \rangle \) with \( S_{\lambda} = S_0 - \{X \upharpoonright GR(\lambda)(X) \in \{0, \frac{1}{2}\}\} \). Let \( \lambda_1 = \lambda \upharpoonright S_{\lambda} \). Consider \( GR(\lambda_1, A_1) \). If it is the same as \( GR(\lambda, A_0) \), then stop and let \( GRB(\lambda, A_0) \) be this complete extension.
step 3): Repeat step 2 on $A_1$.

step 4): Stop when there is no change.

5 Comparisons with other work

This section compares our framework with other techniques that deal with initial values. Our discussions so far and the use of the Gabbay-Rodrigues Iteration Schema, was in the context in which we have an equational approach to the argumentation network and we are given some initial values and we are looking for a “close” equational solution to the initial values.

In [7] proposed two concepts which are directly related to the work presented in this paper. They are both related to the problem of finding an extension given an initial labelling of the arguments in an argumentation network. Their procedure proceeds in two steps. Firstly, the calculate the downward-admissible labelling of the original labelling, which consists of an admissible labelling whose crisp part is maximal under set inclusion of the original labelling. This is done by a procedure which at each step, turns an illegally labelled argument from $\text{in}$ or $\text{out}$ into $\text{und}$ until no illegal crisp values remain. This is called a contraction sequence and is similar to what we do with our sequence of values until it reaches the stable point, except that at each iteration our procedure may contract more than one node simultaneously, whereas theirs only contracts each node per iteration and 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. Even though this search can be optimised, it renders the overall cost of the procedure higher than ours, which is truly bounded by $|S|$. Now given an admissible labelling, a complete extension is constructed by turning illegaly undecided nodes into $\text{in}$ or $\text{out}$ as appropriate. This is called an expansion and the counterpart in our procedure is done by calculating the limit values of the sequence. Obviously, we can only approximate the values and in our examples, we reach the values after which we can no longer guarantee accuracy of the calculation without introducing rounding errors in linear time too (see Figure 9).

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.

We stress that neither are we limited to the discreet values out, in and und, nor to the $E_{d_{\text{max}}}$ equation used in the iteration schema and this allows the application of the schema in the calculation of extensions given different semantics (see Section 7).

One can take a different approach of the above, especially if one is not using any equations. One can take the view that given a network with initial values, we should give an iteration formula that will stabilise on some limit final values. This approach is a bit risky. One needs to explain where the initial values come from and what is the meaning of the iteration formula. One also needs to check whether or not the iteration formula is sound relative to the network's extensions in Dung's sense. In other words, if the initial values correspond to an acceptable Dung extension, does the iteration formula yield a result which does not correspond to a Dung extension? We begin with the work of Pereira et al. [9], which does not take any equational approach but simply iterates on the values of the nodes. We examine in detail what they do.

In what follows, $(S, R)$ is an acyclic argumentation network and $f : S \rightarrow U$ is a function assigning initial values to the nodes in $S$.

**Definition 5.1** Consider the sequence $\alpha_0(X), \alpha_1(X), \ldots, \alpha_i(X), \ldots$, where

\begin{align*}
\alpha_0(X) &= f(X) \\
\alpha_i(X) &= \alpha_{i-1}(X) + \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_{i-1}(Y)\}
\end{align*}

and let

$$\alpha(X) = \lim_{i \to \infty} \frac{1}{2}\alpha_i + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y)\}$$

**Definition 5.2** The attack depth of a node $X$ of an acyclic argumentation network, in symbols $a$-depth$(X)$, is defined recursively as

$$a\text{-depth}(X) = \begin{cases} 
0, & \text{if } \text{Att}(X) = \emptyset \\
\max_{Y \in \text{Att}(X)} a\text{-depth}(Y) + 1, & \text{otherwise}
\end{cases}$$

The function $a$-depth is well-defined, because there are no cycles in $(S, R)$.

**Definition 5.3** Given initial values for the nodes of an acyclic network, the function $\beta : S \rightarrow U$ provides a means of calculating fixed-point values for all nodes as follows.

$$\beta(X) = \begin{cases} 
f(X), & \text{if } a\text{-depth}(X) = 0 \\
\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}, & \text{otherwise}
\end{cases}$$

**Theorem 5.1** $\alpha(X) = \beta(X)$ for all $X \in S$.

**Proof.** The proof is done by induction on the depth of a node.
Base cases: (depth 0) Let $X$ be an argument node of depth 0. By definition, $X$ has no attacks. It follows that

$$
\alpha_0(X) = f(X)
$$
$$
\alpha_1(X) = \frac{1}{2} \alpha_0(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\}
$$
$$
= \frac{1}{2} f(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\}
$$
$$
= f(X)
$$
$$
\alpha_2(X) = \frac{1}{2} \alpha_1(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y)\}
$$
$$
= \frac{1}{2} f(X) + \frac{1}{2} f(X) = f(X)
$$
$$
\alpha(X) = \lim_{i \to \infty} \{\frac{1}{2} \alpha_i + \frac{1}{2} f(X)\}
$$
$$
\alpha(X) = f(X) = \beta(X)
$$

(depth 1) Let $X$ be an argument node of depth 1. By definition, all nodes $Y$ attacking $X$ have depth 0. For all such nodes $f(Y) = \alpha_0(Y) = \alpha_1(Y) = \alpha_i(Y) = \ldots = \alpha(Y) = \beta(Y)$.

$$
\alpha_0(X) = f(X)
$$
$$
\alpha_1(X) = \frac{1}{2} \alpha_0(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\}
$$
$$
= \frac{1}{2} f(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$
$$
\alpha_2(X) = \frac{1}{2} \alpha_1(X) + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\} + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$
$$
= \frac{1}{2^2} f(X) + \frac{1}{2^2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\} + \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$
$$
\alpha_3(X) = \frac{1}{2^2} f(X) + \sum_{i=1}^{t} \frac{1}{2^i} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$
$$
= \frac{1}{2^2} f(X) + (1 - \frac{1}{2^i}) \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$
$$
\alpha(X) = \lim_{i \to \infty} \alpha_i(X)
$$
$$
= \lim_{i \to \infty} \frac{1}{2^i} f(X) + (1 - \frac{1}{2^i}) \frac{1}{2} \min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y)\}
$$

45
\[
\begin{align*}
\alpha_0(X) &= f(X) \\
\alpha_1(X) &= \frac{1}{2}\alpha_0(X) + \frac{1}{2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\} \\
\alpha_2(X) &= \frac{1}{2^2}\alpha_0(X) + \frac{1}{2^2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\} + \\
&\quad \frac{1}{2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y)\} \\
\alpha_3(X) &= \frac{1}{2^3}\alpha_0(X) + \frac{1}{2^3}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\} + \\
&\quad \frac{1}{2^2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y)\} + \\
&\quad \frac{1}{2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_2(Y)\} \\
&= \frac{1}{2^3}\alpha_0(X) + \frac{1}{2^3}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\} + \\
&\quad \frac{1}{2^2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y)\} + \\
&\quad \frac{1}{2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_2(Y)\} \\
\alpha_i(X) &= \frac{1}{2^i}\alpha_0(X) + \frac{1}{2^i}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y)\} + \\
&\quad \frac{1}{2^{i-1}}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y)\} + \ldots + \\
&\quad \frac{1}{2}\min\{f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_{i-1}(Y)\} \\
&= \beta(X)
\end{align*}
\]

Assume that the theorem holds for nodes with attack depth up to \( k \) and let \( X \) be an argument node whose attack depth is \( k + 1 \). We have that
\[ \alpha_{i+1}(X) = \frac{1}{2^i} f(X) + \sum_{i=1}^{\infty} \frac{1}{2^i} \left( \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y) \} \right) \]

\[ = \frac{1}{2^i} f(X) + \left( 1 - \frac{1}{2^i} \right) \left( \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y) \} \right) \]

\[ \alpha(X) = \lim_{i \to \infty} \frac{1}{2^i} f(X) + \left( 1 - \frac{1}{2^i} \right) \left( \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y) \} \right) \]

\[ = \lim_{i \to \infty} \left( 1 - \frac{1}{2^i} \right) \left( \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y) \} \right) \]

\[ = \lim_{i \to \infty} \left( \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_i(Y) \} \right) \]

\[ = \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha(Y) \} \]

\[ \alpha(X) = \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha(Y) \} \]

But the attack depth of the nodes \( Y \in \text{Att}(X) \) is no higher than \( k \). By the induction hypothesis we have that \( \alpha(Y) = \beta(Y) \) for all \( Y \in \text{Att}(X) \) and hence

\[ \alpha(X) = \min \{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \} = \beta(X) \]

The theorem shows that when there are no cycles, for any node \( X \), the sequence \( \alpha_i(X) \) converges to the value \( \beta(X) \), which can be calculated by considering the tree with root \( X \) and propagating values from the leaves to the root according to Definition 5.3.

One can argue that the procedure is not sound with respect to admissibility. In particular, the algorithm does not turn arbitrary initial values into admissible ones. If we give initial value 0 to a node which should not be labelled out, the algorithm does not correct the node’s value and it remains illegally out. Likewise, if we start with a two-node cycle \( A \leftrightarrow B \) and provide initial values to \( A \) and \( B \) that correspond to a complete extension, say \( A = 1, B = 0 \), in the limit we get values \( A = \frac{1}{2} \) and \( B = 0 \). Ideally, the initial values should remain the same as in the Gabbay-Rodrigues Iteration Schema (and indeed Caminada and Pigozzi’s down-admissible/up-complete construction).

6 General Methodology and Discussion

Suppose you start with a large network with cycles. Rewrite it into an acyclic network of strongly connected components of the initial network. Look at the direction of attacks at take the terminal components in the acyclic network. Use initial values of your choice for each of the nodes in each cycle (for example, for each cycle, choose a maximum conflict-free set and assign the value 1 members of the set and 0 to its complement). Apply the Gabbay-Rodrigues formula to get the unique maximal downward admissible extension. Extend it to a unique extension, if possible. For each non-terminal component, recompute the strongly connected components and repeat. At the end we get an extension.
The Gabbay-Rodrigues formula preserves original complete extensions within components. If the initial values are legitimate, they correspond to an extension which will be preserved in the final result (this will not be the case with Villata). If they are not, we will get the maximal admissible subset.

6.1 Reasoning with initial values

In Section 1, we suggested that one of the possible ways of dealing with initial values was to incorporate them in the formulation of the equations. We argued that since terminal nodes are not affected by other nodes, their equilibrium values should correspond to the initial values assigned to them. This would be straightforward to implement, except that we also have to deal with non-terminal nodes. However, the treatment of terminal nodes gives us a clue on how to deal with the initial values of nodes that are involved in loops. The idea is to think of the initial value of a node as the equilibrium value of a node that supports it. Consider again the argumentation network shown in Figure 1 and suppose the initial values for $X$, $Y$ and $Z$ are $V_0(X)$, $V_0(Y)$ and $V_0(Z)$, respectively. Using $E_{\text{max}}$ alone, we would obtain the system of equations

\[
\begin{align*}
V_e(X) &= 1 \\
V_e(Y) &= 1 - \max\{V_e(X)\} = 0 \\
V_e(Z) &= 1 - \max\{V_e(Y)\} = 1
\end{align*}
\]

and thus we would not get the desired result $V_e(X) = V_0(X)$.

Now consider the extension $N' = \langle S', R' \rangle$ of the original network $N = \langle S, R \rangle$ constructed by adding to $S$ two shadow nodes $X_0$ and $X_0'$ for each $X \in S$, and adding to $R$ the edges $(X_0, X_0)$, $(X_0, X_0')$ and $(X_0, X)$ (see Figure 11). $N'$ will

![Diagram of an extended network allowing for arbitrary initial values for the nodes in the network of Figure 1](image)

Figure 11: An extended network allowing for arbitrary initial values for the nodes in the network of Figure 1
generate the following equations

\[ V_e(X_0) = 1 - V_e(X'_0) \]
\[ V_e(X'_0) = 1 - V_e(X_0) \]
\[ V_e(X) = 1 - V_e(X'_0) \]
\[ V_e(Y_0) = 1 - V_e(Y'_0) \]
\[ V_e(Y'_0) = 1 - V_e(Y_0) \]
\[ V_e(Y) = 1 - \max\{V_e(Y_0), V_e(X)\} \]
\[ V_e(Z_0) = 1 - V_e(Z'_0) \]
\[ V_e(Z'_0) = 1 - V_e(Z_0) \]
\[ V_e(Z) = 1 - \max\{V_e(Z_0), V_e(Y)\} = 1 \]

and thus will accept any solution in which the values of \( X_0 \) and \( X'_0 \) are complementary. The solution in which the value \( V_e(X_0) \) is \( 1 - \), will give \( V_e(X) = 1 - (1 - V_0(X)) = V_0(X) \), and hence there is a solution in the original network in which \( V_e(X) = V_0(X) \).

7 Conclusions and Future Research

This paper investigated aspects concerned with argumentation networks where the arguments are provided with initial values. We are aware that assigning values to nodes and propagating values through the network has been independently investigated before as in, e.g., [8, 2]. 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 naturally using equations. In addition, there are many mathematical tools to help find solutions to the equations.

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 Henry Prakken [19]; or they can be obtained as the result of the merging of several networks, as proposed in [16, 15].

In this paper, we showed that the equations can be solved through an iterative process, as in Newton’s method and as such one can regard initial values as initial guesses or a desired configuration of the extension. The Gabbay-Rodrigues Iteration Schema takes the following generalised form:

\[ V_{i+1}(X) = (1 - V_i(X)) \cdot \min\{1/2, g(N(X))\} + V_i(X) \cdot \max\{1/2, g(N(X))\} \]
In this paper, we considered the special case where \( g \) is \( \min \) and \( \mathcal{N}(X) \) is the set of complemented values of the nodes in the “neighbourhood” of \( X \) (i.e., the attackers of \( X \)). Other operations can be used for argumentation systems, whose relationship with the schema is being further investigated. One such operation is \textit{product}, which unlike \( \min \) combines the strength of the attacks on a node. Another interesting possibility is to use the schema for \textit{abstract dialectical frameworks} (ADFs) \cite{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 \( R = \{(a, b), (b, c), (c, c)\} \). The ADF equations are: \( C_a = \top, C_b = a, C_c = c \land b \) 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 \( m_3 = (t, t, f, u) \). The Gabbay-Rodrigues 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 \( \min \), we showed that the values generated at each iteration in the schema eventually “stabilise” by changing illegal crisp values into undecided. This process will calculate the down-admissible labelling of the initial values, as in \cite{7}, in time linear to the set of arguments \((t \leq |S|)\). If we carry on the calculation, the values of the sequence in the limit will correspond to a complete extension of the original network. As a consequence, if values corresponding to a legitimate extension are given as input, then the sequence will immediately stabilise at those values. In practice, a few iterations are sufficient to indicate what the values will converge to in the limit. We have also outlined a procedure which can improve on the calculation above by propagating crisp values and replacing the remaining undecided values with their initial counterparts after each run of the iterations. This procedure terminates when no new crisp values are generated. Original crisp values which are compatible with a calculated extension are preserved in this way and hence we can end up with a larger complete extension than the one obtained in a single run. This extension is as compatible as possible with the initial values.

There is a wealth of research to be done using this approach in the investigation of semantics of argumentation systems. We have already established minimal conditions operations for argumentation systems must satisfy if they are to be used in the equations. For reasons of space, these could not be presented here. In future work, we will present this in richer detail.

Acknowledgements

The authors would like to thank Martin Caminada and Sanjay Modgil for comments and discussions on the topic of this paper.

\footnote{Note that \( 1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} = \min_{Y \in \text{Att}(X)} \{1 - V(Y)\} \).}
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 the 12th International Conference on the Principles of Knowledge Representation and Reasoning: KR’10, pages 102 – 111. AAAI Press, 2010.

[4] M. Caminada. An algorithm for computing semi-stable semantics. In Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU ’07, pages 222–234, Berlin, Heidelberg, 2007. Springer-Verlag.

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

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

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

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

[9] C. da Costa Pereira, A.G.B. Tettamanzi, and S. Villata. Changing one’s mind: erase or rewind? possibilistic belief revision with fuzzy argumentation based on trust. In Proceedings of the 22nd International joint conference on artificial intelligence : IJCAI’11, pages 164 – 171, Menlo Park, 2011. AAAI Press.

[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] S. Egelmez, J. Leite, and J. Martins. Extending social abstract argumentation with votes on attacks. In Proceedings of the 2nd International Workshop on Theory and Applications of Formal Argumentation (TAFA’13), to appear 2014.

[12] D. M. Gabbay. Introducing equational semantics for argumentation networks. DOI: 10.1007/978-3-642-22152-1_2, 2011.

[13] D. M. Gabbay. Equational approach to argumentation networks. Argument and Computation, 3:87–142, 2012. DOI: 10.1080/19462166.2012.704398.
[14] D. M Gabbay. *Meta-logical Investigations in Argumentation Networks*, volume 44 of *Studies in Logic: Mathematical Logic and Foundations*. College Publications, 2013. ISBN: 978-1-84890-103-2.

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

[16] D. M. Gabbay and O. Rodrigues. A numerical approach to the merging of argumentation networks. In M. Fisher, L. van der Torre, M. Dastani, and G. Governatori, editors, *Proceedings of CLIMA XIII*, pages 195–212. Springer-Verlag, 2012.

[17] J. Leite and J. Martins. Social abstract argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011. To appear.

[18] Newton’s method. [http://en.wikipedia.org/wiki/Newton](http://en.wikipedia.org/wiki/Newton)

[19] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
A  Predator-Prey and Argumentation Motivating Case Studies

Let us motivate our ideas through two main examples. Our purpose is to make some conceptual distinction about iteration processes.

Example A.1 Let us look at another example from biology. This is a model by M. P. Hassell (1978) of two parasitoids (P and Q) and one host (N) model. The ecology is described in Figure 12. The equations are (see [1], p. 295)

\[
\begin{align*}
N_{t+1} &= \lambda N_t f_1(P_t) f_2(Q_t) \\
P_{t+1} &= N_t [1 - f_1(P_t)] \\
Q_{t+1} &= N_t f_1(P_t) [1 - f_2(Q_t)]
\end{align*}
\]

where \(N, P\) and \(Q\) denote the host and two parasitoid species in generations \(t\) and \(t + 1\), \(\lambda\) is the finite host rate of increase and the functions \(f_1\) and \(f_2\) are the probabilities of a host not being found by \(P_t\) or \(Q_t\) parasitoids, respectively. This model applies to two quite distinct types of interaction that are frequently found in real systems. It applies to cases where \(P\) acts first, to be followed by \(Q\) acting only on the survivors. Such is the case where a host population with discrete generations is parasitized at different developmental stages. In addition, it applies to cases where both \(P\) and \(Q\) act together on the same host stage, but the larvae of \(P\) always out-compete those of \(Q\), should multi-parasitism occur.

The functions \(f_1\) and \(f_2\) are:

\[
\begin{align*}
f_1(P_t) &= \left[ 1 + \frac{a_1 P_t}{k_1} \right]^{-k_1} \\
f_2(Q_t) &= \left[ 1 + \frac{a_2 Q_t}{k_2} \right]^{-k_2}
\end{align*}
\]

where \(k_1\) and \(k_2\), \(a_1\) and \(a_2\) are constants.

To simplify and later compare the biological model with the argumentation model, we put \(k_1 = k_2 = -1\).

Figure 12: A sample biological network.

This gives

\[
\begin{align*}
f_1(P_t) &= 1 - a_1 P_t \\
f_2(Q_t) &= 1 - a_2 Q_t
\end{align*}
\]
and therefore, the equations are
(1, t): \( N_{t+1} = \lambda N_t (1 - a_1 P_t)(1 - a_2 Q_t) \)
(2, t): \( P_{t+1} = a_1 N_t P_t \)
(3, t): \( Q_{t+1} = a_2 Q_t N_t (1 - a_1 P_t) \)

At a state of equilibrium, we get the following fixpoint equations:

\[
\begin{align*}
N &= \lambda N (1 - a_1 P)(1 - a_2 Q) \quad \text{(5)} \\
P &= a_1 N P \quad \text{(6)} \\
Q &= a_2 Q N (1 - a_1 P) \quad \text{(7)}
\end{align*}
\]

Ignoring the zero solution \( P = Q = N = 0 \), we get from (6) that

\[
N = \frac{1}{a_1} \quad \text{(9)}
\]

and from (7) we get

\[
1 = a_2 \cdot \frac{1}{a_1} (1 - a_1 P) \quad \text{(10)}
\]

and hence

\[
\begin{align*}
a_1 &= a_2 - a_2 a_1 P \\
P &= \frac{a_2 - a_1}{a_1 a_2}
\end{align*}
\]

From (5), we get

\[
\begin{align*}
1 &= \lambda \left(1 - \frac{a_1(a_2 - a_1)}{a_1 a_2}\right)(1 - a_2 Q) \\
1 &= \frac{\lambda a_1}{a_2} (1 - a_2 Q)
\end{align*}
\]

so

\[
\begin{align*}
\frac{a_2}{\lambda a_1} &= 1 - a_2 Q \\
a_2 Q &= \frac{\lambda a_1 - a_2}{\lambda a_1} \\
Q &= \frac{\lambda a_1 - a_2}{\lambda a_1 a_2}
\end{align*}
\]

To have a specific example for discussion let \( a_1 = 2, a_2 = 3, \lambda = 2 \). We get \( N = 0.5, P = \frac{1}{6} \) and \( Q = \frac{1}{12} \). Indeed, substituting these values in the equations we have

\[
(1) \quad 1 = 2(1 - 2 \cdot \frac{1}{6})(1 - \frac{3}{12}) = 2 \cdot \frac{2}{3} \cdot \frac{3}{12} = 2 \cdot \frac{18}{36} = 1
\]
(2) \[ 1 = 2 \cdot \frac{1}{2} = 1 \]

(3) \[ 1 = 3 \cdot \frac{1}{2}(1 - \frac{2}{6}) = \frac{3 \cdot \frac{4}{6}}{2} = 1 \]

Let us substitute \( a_1, a_2 \) and \( \lambda \) in the equations and pretend we do not know the solution. We get the equations

(1*) \[ N = 2N(1 - 2P)(1 - 3Q) \]
(2*) \[ P = 2PN \]
(3*) \[ Q = \frac{3}{2}Q(1 - 2P) \]

So we have a system of equations modelling a certain ecology.

The equations above give rise to the iteration equations

(1*,i): \[ N_{i+1} = 2\lambda N_i(1 - 2P_i)(1 - 3Q_i) \]
(2*,i): \[ P_{i+1} = 2N_iP_i \]
(3*,i): \[ Q_{i+1} = \frac{3}{2}Q_iN_i(1 - 2P_i) \]

Let us discuss our options. We have a system of equations involving \( N, P \) and \( Q \) and we want to solve it. We do not know whether there are solutions. Option 1, a mathematical view: Let us just find a solution. We can guess a candidate solution, use Newton-Raphson’s method and iterate. Let us do this with the guess \( N_0 = P_0 = Q_0 = \frac{1}{2} \) and iterate. These are equations (1*,i), (2*,i) and (3*,i) for \( i = 1 \).

Because the equations come from ecological considerations, the iterations are not just a numerical device but also have an evolutionary meaning. However, our view is purely mathematical. The corresponding to the meaning is accidental.

We get

\[ \begin{align*}
N_1 &= 2 \cdot \frac{1}{2} \cdot N_0(1 - 1)(1 - \frac{2}{3}) = 0 \\
P_1 &= 2 \cdot \frac{4}{7} \cdot \frac{1}{2} = 0 \\
Q_1 &= \frac{3}{2} \cdot Q_0N_0(1 - 2P_0) = 0 \\
N_2 &= 0 \\
P_2 &= 0 \\
Q_2 &= 0 \\
\end{align*} \]

We converge to the all zero solution.

Option 2, a semantical view. We seek a solution motivated not by mathematics but by the meaning of the equations; by ecological considerations. So let us adopt the friends of parasites view and say that we are equal and we all have a right to live and so let us seek a steady state of compromise and living together in tolerance and understanding, namely \( N_0 = P_0 = Q_0 = \frac{1}{2} \).

Unfortunately using Newton-Raphson’s iteration formula leads us, as shown above, to the solution \( P = Q = N = 0 \). In biological terms this is not good, it means everything is dead. So we may need a better iteration formula, a formula suitable for the biological interpretation.

We can choose to be selfish and cruel and start with \( N_0 = 1 \) and \( P_0 = Q_0 = 0 \). This means we aim at full population and no parasites. Interating the equations will give us
\[ N_1 = 2 \]
\[ P_1 = 0 \]
\[ Q_1 = 0 \]
\[ N_k = 2^k \]
\[ P_k = 0 \]
\[ Q_k = 0 \]

This does not lead to a solution. It diverges!

Let us now choose other values, perhaps \( N_0 = 0.9 \) and \( P_0 = Q_0 = 0.1 \), which comes close to the selfish proposal. Iterating gives us

\[ N_1 = 2 \times \frac{9}{10} \times (1 - 0.2)(1 - 0.3) = 1.008 \]
\[ P_1 = 2 \times 0.1 \times 0.9 = 0.18 \]
\[ Q_1 = \frac{3}{2} \times 0.1 \times 0.8 = 0.12 \]

\[ N_2 = 2 \times 1.008 \times (1 - 0.36)(1 - 0.36) = 0.8257536 \]
\[ P_2 = 2 \times 0.18 \times 1.008 = 0.36288 \]
\[ Q_2 = \frac{3}{2} \times 0.12 \times 0.64 = 0.1152 \]

We can carry on, we might converge to the solution \( N = \frac{1}{2}, P = \frac{1}{6} \) and \( Q = \frac{1}{12} \). CHECK!!!

**Remark A.1** The conclusion we draw from Example [A.1] is that we must be aware that some iteration processes can be mathematical only, just possibly leading to a mathematical solution but otherwise semantically meaningless, and some may be semantically meaningful and useful in the context of the application area from which the equations arise.

This observation shall become sharper and clearer in the case of our next example from abstract argumentation.

**Example A.2** Consider Figure [12] again but this time as an argumentation network where \( N, P, Q \) are arguments. This network has three extensions \( E_1, E_2, E_3 \), namely

\[ E_1 = P = 1 \ (P \text{ is “in”}) \]
\[ N = Q = 0 \ (N \text{ and } Q \text{ are “out”}) \]

\[ E_2 = N = 1 \ (N \text{ is “in”}) \]
\[ P = Q = 0 \ (P \text{ and } Q \text{ are “out”}) \]

\[ E_3 = N = P = Q = \frac{1}{2} \]
\[ = \text{ all undecided} \]

We provide semantics for abstract argumentation in terms of equations (see [12, 13, 14]). There are many options for possible equations. We introduce two of them here \( \text{Eq}_{\text{max}} \) and \( \text{Eq}_{\text{inv}} \).

Let \( \gamma(X) = \{Y_1, \ldots, Y_k\} \) be all the attackers of \( X \). Consider \( X, Y_1, \ldots, Y_k \) as variables ranging over \([0, 1]\). Write
The equation we write is

(*) \( X = G(\gamma(X)) \)

where \( G \) can be \( G_{\text{max}} \) or \( G_{\text{inv}} \) or some other function. A complete extension is any solution to the system of equations \( X = G(\gamma(X)) \), for all arguments \( X \). We consider \( X = 1 \) to mean \( X \) is “in” and \( X = 0 \) to mean \( X \) is “out”. For the case of \( G_{\text{max}} \), the complete extensions coincide with the traditional Dung complete extensions, where in addition to the “in”/“out” values above, the value \( 0 < X < 1 \) means \( X \) is undecided.

The system with \( G_{\text{inv}} \) is sensitive to the number of attackers of a node. For example, assume there are 10 undecided attackers \( Y_i \) of \( X \) each having value \( \frac{1}{2} \) (undecided), then the value of \( X \) becomes \( \frac{1}{10} \) under \( G_{\text{inv}} \), while in the traditional Dung approach (\( G_{\text{max}} \)), the value of \( X \) is simply \( \frac{1}{2} \). \( X \) is nearer to 0 (i.e., “out”) in the \( G_{\text{inv}} \) case!

The \( G_{\text{inv}} \) equations for the network in Figure 12 are

(i) \( N = (1 - P)(1 - Q) = G_{\text{inv}}(N) \)
(ii) \( P = (1 - N) = G_{\text{inv}}(P) \)
(iii) \( Q = (1 - P)(1 - N) = G_{\text{inv}}(Q) \)

The background material on the equational approach is given in the next section. It is sufficient to say here that a “complete extension” is any solution of these equations for the “variables” \( N, P \) and \( Q \). There are two solutions in the \( G_{\text{inv}} \) case: \( N = 1, P = Q = 0 \) and \( N = 0, P = 1 \) and \( Q = 0 \). The extension “all undecided” restricted to \( N = P = Q = \frac{1}{2} \) is not possible. These values do not solve the \( G_{\text{inv}} \) equations.

However, if we understand “undecided” as any value in \((0, 1)\), then there may well be a solution!

The values \( N = P = Q = \frac{1}{2} \) do solve the \( G_{\text{max}} \) equations.

1. If we think in terms of option 1), and just seek a mathematical solution to the equations using Newton-Raphson’s iteration method, we will get the solution \( P = 1, N = Q = 0 \)

2. The only time we get the other solution \( N = 1, P = Q = 0 \) is that we guess it.

Thus if we think in terms of semantics, the Newton-Raphson process is not adequate. We need a new iteration process, which if we start very near the solution \( N = 1, P = Q = 0 \) (say \( N = -.9, P = Q = 0.1 \)), then we get convergence to the desired solution.

We now have similar two options for this argumentation case, as we had in the previous ecological example. In Option 1 – the mathematical option, we just want to find a solution to the equations. This is a numerical analysis problem.
There are many iteration formulae which start with a candidate possible solution and iterates in the hope of converging to a solution.

Option 2 is motivated by semantics. The initial values (candidate solution) come from the application area where the argumentation network is being used and we require a solution to the equations (i.e., an extension) which reflects (being near?) to these values. Here we cannot accept any solution. we want solutions which reflect the input. So we need to devise iteration formulas which have a semanical meaning. In addition to the usual mathematical properties that the iteration sequence converges. This point is important. Suppose we have 100 voters who vote on what arguments they support. 90 vote for the extension \( N = 1, P = Q = 0 \) and then vote for \( N = 0, P = 1 \) and \( Q = 0 \). We thus say that we have a numerical assignment \( N = 0.9, P = 0.1 \) and \( Q = 0.1 \). We now ask what extension is nearest to this majority vote?

We need an iteration formula which will start with the initial values \( N = 0.9, P = 0.1 \) and \( Q = 0.1 \) and converges to \( N = 1, P = 0 \) and \( Q = 0 \). We do not want convergence to some far away value. Remember, we want an interation process which respects the meaning of these numbers (i.e., they come from votes). We do not want to compute all extensions and select the nearest one!

Let us see if the Newton-Raphson's iteration method works for this case. We start by taking initial values \( N_0 = 1 \epsilon, P_0 = 0 \) and \( Q_0 = 0 \) and let us iterate then iterating, first for the case of \( G_{max} \) (equations (m), (mm) and (mmm)) and then we check for \( G_{inv} \) (equations (i), (ii) and (iii)). We shall see that these iteration formulae are not satisfactory, i.e., they do not work.

**case \( G_{max} \) (equations (m)--(mmm))**

Initial values \( N_0 = 1 - \epsilon, P = 0 \) and \( Q = 0 \). We get

\[
\begin{align*}
N_1 &= 1, & P_1 &= \epsilon, & Q_1 &= \epsilon \\
N_2 &= 1 - \epsilon, & P_2 &= \epsilon, & Q_2 &= 0 \\
N_3 &= 1 - \epsilon, & P_3 &= \epsilon, & Q_3 &= \epsilon \\
N_4 &= 1 - \epsilon, & P_4 &= \epsilon, & Q_4 &= \epsilon
\end{align*}
\]

There is convergence here, but since \( \epsilon \) is arbitrary, this is not satisfactory.

We want \( N = 1, P = Q = 0 \).

**case \( G_{inv} \) (equations (i)--(iii))**

Initial values \( N_0 = 1 - \epsilon, P = 0 \) and \( Q = 0 \). We get

\[
\begin{align*}
N_1 &= 1, & P_1 &= \epsilon, & Q_1 &= \epsilon \\
N_2 &= 1 - (1 - \epsilon)^2, & P_2 &= 0, & Q_2 &= 0 \\
N_3 &= 1, & P_3 &= 1 - (1 - \epsilon)^2, & Q_3 &= 1 - (1 - \epsilon)^2 \\
N_4 &= (1 - \epsilon)^4, & P_4 &= 0, & Q_4 &= 0
\end{align*}
\]

We are getting nowhere. Let us show by induction that:

\[
N_{2k+1} = 1, \quad P_{2k+1} = 1 - (1 - \epsilon)^{2^k}, \quad Q_{2k+1} = 1 - (1 - \epsilon)^{2^k}
\]

\[
N_{2k} = (1 - \epsilon)^{2^k}, \quad P_{2k} = 0, \quad Q_{2k} = 0
\]

Assume the above values for \( m = 2k \). Show we get the right inductive values for \( m + 1 \).
\[ N_{2k+1} = (1 - P_{2k})(1 - Q_k) \]
\[ = (1 - 0)(1 - 0) \]
\[ = 1 \]
\[ P_{2k+1} = (1 - N_{2k}) \]
\[ = 1 - (1 - \epsilon)^{2^k} \]
\[ Q_{2k+1} = (1 - P_{2k})(1 - N_{2k}) \]
\[ = 1 - (1 - \epsilon)^{2^k} \]

Assume the above values for \( m = 2k + 1 \). Show the same for \( m + 1 \).

\[ N_{2(k+1)} = (1 - P_{2k+1})(1 - Q_{2k+1}) \]
\[ = (1 - \epsilon)^{2^k} (1 - \epsilon)^{2^k} \]
\[ = (1 - \epsilon)^{2^{k+2}} \]
\[ = (1 - \epsilon)^{2^{k+1}} \]

We see that the \( 2k + 1 \) items (subsequence) converge to \( N = 1, P = 1 \), and \( Q = 1 \) and the \( 2^k \) items (subsequence) converge to \( N = 0, P = 0 \), and \( Q = 0 \). In other words, we get nothing.

We see that the iteration formulae (i)–(iii) and (m)–(mmm) are not suitable for preserving and respecting the semantical meaning. The initial values \( N_0 = 1 - \epsilon, P = Q = 0 \), come from voting and when we apply the Newton iteration process we do not get semantically meaningful answers. We need to devise our own iteration formula, or maybe several such formulae, tailored for different types of application areas.

Let us now introduce the Gabbay-Rodrigues iteration formula, the main subject matter of this paper. Let \( (S, R) \) be an argumentation network and \( X, Y \in S \) be considered variables and let the equations be \( X = G(\lambda(X)) \), where \( G \) can be either \( G_{\text{max}} \) or \( G_{\text{inv}} \). Let \( V_i(X) \) be the iteration value that our formula gives to \( X \) at iteration step \( i \). Then our formula is

\[ V_{i+1}(X) = (1 - V_i(X)) \cdot \min\{\frac{1}{2}, G(\{V_i(Y_j)\})\} + V_i(X) \cdot \max\{\frac{1}{2}, G(\{V_i(Y_j)\})\} \]
So for our example and $G_{\text{max}}$ we get

\[
V_{i+1}(N) = (1 - V_i(N)) \cdot \min\{\frac{1}{2}, 1 - \max\{V_i(P), V_i(Q)\}\} + V_i(N) \cdot \max\{\frac{1}{2}, 1 - \max\{V_i(P), V_i(Q)\}\}
\]

\[
V_{i+1}(P) = (1 - V_i(P)) \cdot \min\{\frac{1}{2}, 1 - V_i(N)\} + V_i(P) \cdot \max\{\frac{1}{2}, 1 - V_i(N)\}
\]

\[
V_{i+1}(Q) = (1 - V_i(Q)) \cdot \min\{\frac{1}{2}, 1 - \max\{V_i(P), V_i(N)\}\} + V_i(Q) \cdot \max\{\frac{1}{2}, 1 - \max\{V_i(P), V_i(N)\}\}
\]

For the case of $G_{\text{inv}}$, we get

\[
V_{i+1}(N) = (1 - V_i(N)) \cdot \min\{\frac{1}{2}, (1 - V_i(P))(1 - V_i(Q))\} + V_i(N) \cdot \max\{\frac{1}{2}, (1 - V_i(P))(1 - V_i(Q))\}
\]

\[
V_{i+1}(P) = (1 - V_i(P)) \cdot \min\{\frac{1}{2}, 1 - V_i(N)\} + V_i(P) \cdot \max\{\frac{1}{2}, 1 - V_i(N)\}
\]

\[
V_{i+1}(Q) = (1 - V_i(Q)) \cdot \min\{\frac{1}{2}, (1 - V_i(P))(1 - V_i(N))\} + V_i(Q) \cdot \max\{\frac{1}{2}, (1 - V_i(P))(1 - V_i(N))\}
\]

Let us now take the initial conditions $V_0(N) = 1 - \epsilon$, $V_0(P) = 0$ and $V_0(Q) = 0$ and calculate the iterations according to the $G_{\text{max}}$ case with $\epsilon = 0.001$. The values will converge to $\frac{1}{2}$.

We now turn to the $G_{\text{inv}}$ case. Start with $V_0(N) = 1 - \epsilon$, $V_0(P) = 0$. We
have

\[
V_1(N) = (1 - V_0(N)) \cdot \min\left\{ \frac{1}{2}, (1 - V_0(P))(1 - V_0(Q)) \right\} + \\
V_0(N) \cdot \max\left\{ \frac{1}{2}, (1 - V_0(P))(1 - V_0(Q)) \right\}
\]

\[
= \epsilon \cdot \min\left\{ \frac{1}{2}, 1 \right\} + (1 - \epsilon) \cdot 1
\]

\[
= \frac{\epsilon}{2} + 1 - \epsilon
\]

\[
= 1 - \frac{\epsilon}{2}
\]

\[
V_1(P) = (1 - V_0(P)) \cdot \min\left\{ \frac{1}{2}, 1 - V_0(N) \right\} + \\
V_0(P) \cdot \max\left\{ \frac{1}{2}, 1 - V_0(N) \right\}
\]

\[
= \min\left\{ \frac{1}{2}, \epsilon \right\}
\]

\[
= \epsilon
\]

\[
V_1(Q) = (1 - V_0(Q)) \cdot \min\left\{ \frac{1}{2}, (1 - V_0(P))(1 - V_0(N)) \right\} + \\
V_0(Q) \cdot \max\left\{ \frac{1}{2}, (1 - V_0(P))(1 - V_0(N)) \right\}
\]

\[
= \min\left\{ \frac{1}{2}, \epsilon \right\}
\]

\[
= \epsilon
\]

We continue

\[
V_2(N) = \frac{\epsilon}{2} \cdot \min\left\{ \frac{1}{2}, (1 - \epsilon)^2 \right\} + (1 - \frac{\epsilon}{2}) \cdot \max\left\{ \frac{1}{2}, (1 - \epsilon)^2 \right\}
\]

\[
= \frac{\epsilon}{4} + (1 - \frac{\epsilon}{2})(1 - \epsilon)^2
\]

We need to approximate if we want to continue by hand.

\[
V_2(N) = \frac{\epsilon}{4} + (1 - \frac{\epsilon}{2})(1 - 2\epsilon + \epsilon^2)
\]

\[
= \frac{\epsilon}{4} + (1 - 2\epsilon + \epsilon^2) - \frac{\epsilon}{2} + \epsilon^2 - \frac{\epsilon^3}{2}
\]

Neglect \(\epsilon^2, \epsilon^3, \text{ etc.}\)

\[
N \approx \frac{\epsilon}{4} + 1 - 2\epsilon - \frac{\epsilon}{2}
\]

\[
\approx 1 - \frac{9}{4}\epsilon
\]

61
\[ V_2(P) = (1 - \epsilon) \cdot \min\left\{ \frac{1}{2}, 1 - V_1(N) \right\} + \epsilon \max\left\{ \frac{1}{2}, 1 - V_1(N) \right\} \]
\[ = \frac{(1 - \epsilon)}{2} \cdot \frac{\epsilon}{2} + \frac{\epsilon}{2} \]
\[ = \frac{\epsilon}{4} - \frac{\epsilon^2}{4} + \frac{\epsilon}{2} \]
\[ = \frac{3}{4} \epsilon - \frac{\epsilon^2}{4} \]

\[ V_2(Q) = (1 - \epsilon) \cdot \min\left\{ \frac{1}{2}, (1 - V_1(P))(1 - V_1(N)) \right\} + \epsilon \max\left\{ \frac{1}{2}, (1 - V_1(P))(1 - V_1(N)) \right\} \]
\[ = (1 - \epsilon) \cdot \min\left\{ \frac{1}{2}, (1 - \epsilon)\frac{\epsilon}{2} \right\} + \epsilon \max\left\{ \frac{1}{2}, (1 - \epsilon)\frac{\epsilon}{2} \right\} \]
\[ = (1 - \epsilon)^2 \cdot \frac{\epsilon}{2} + \frac{\epsilon}{2} \]
\[ = \frac{\epsilon}{2}(1 - \epsilon)^2 + 1 \]
\[ = \frac{\epsilon}{2}(1 - 2\epsilon + \epsilon^2 + 1) \]
\[ = \frac{\epsilon}{2} - \frac{2\epsilon^2}{2} + \frac{\epsilon^3}{2} + \frac{\epsilon}{2} \]

We need to approximate and neglect \( \epsilon^2, \epsilon^3, \) etc. We get \( Q \approx \epsilon. \)

In summary, we get

\[ N \approx 1 - \frac{9}{4} \epsilon \]
\[ P \approx \frac{3}{4} \epsilon \]
\[ Q \approx \epsilon \]

Let us conclude this Appendix by saying a bit more about the equational approach. The equational approach to an abstract argumentation system views an argumentation graph \( \langle S, R \rangle \) as a generator of equations for functions \( f : S \rightarrow [0,1] \). Any function \( f \) which is a solution to the equations is considered a complete numerical extension for the original graph and for every argument \( X \in S \), we can give the interpretation

\( f(X) = 1 \) means that \( X \) is definitely “in”
\( f(X) = 0 \) means that \( X \) is definitely “out”
\( f(X) \in (0,1) \) indicates a certain degree of undecidedness about \( X \)

The details of this have been previously worked out in a series of papers [1, 12, 13, 16, 15, 14]. Using the above as a starting point, we address in this
paper the following question. Suppose we are given an initial numerical function \( V_0 : S \rightarrow [0, 1] \), which 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. This function may not satisfy the equations. Whatever its origins, we are looking for methods for finding the “closest” compatible solution satisfying the equations.

The reader should note that the same problem arises in traditional three-valued argumentation systems. In a numerical context, there is a wide range of mathematical methods available to find a solution. We study the interdependencies between equations and initial values and we propose algorithms to solve the equations.

Furthermore, there is a variety of papers in the literature using numerical values in the form of weights to attacks and/or arguments. We compare our systematic methodology to what they do.

We stress the need for our framework because in many application areas numerical initial values in \( U \) are generated at least in an intermediate stage. It is therefore of practical importance that we know how to deal with such initial values.

B Numerical Argumentation Networks

In [1], the idea of support and attack networks was initially proposed. These networks allow for the assignment of initial values to the nodes of the graph; the specification of a transmission factor associated with the strength with which an attack between arguments is carried out; and the higher-level notion of an attack to an attack. In [10], we showed how some of these features can be used in the merging of argumentation networks. The numerical argumentation networks we now propose share some of the features of the support and attack networks, but introduce a functional approach to the computation of interaction between nodes.

Definition B.1 (Numerical Argumentation Network) A numerical argumentation network is a tuple \( \langle S, R, V_0, V_e, g, h, \Pi \rangle \), where

- \( S \) is a set of nodes, representing arguments;
- \( R \subseteq S^2 \) is an attack relation, where \((X,Y) \in R\) means “\(X\) attacks \(Y\)”;
- \( V_0 : S \rightarrow U \) is a function assigning initial values to the nodes in \( S \);
- \( g \) is a function to combine attacks to a node;
- \( h \) is a function to combine the initial value of a node with the value of its attack;
- \( \Pi \) is an algorithm to compute equilibrium values \( V_e(X) \), for each node \( X \in S \).
We assume that $g$ and $h$ are possibly distinct argumentation-friendly functions according to Definition 1.2. The equilibrium value of a node $X$, $V_e(X)$, is defined as $h(V_0(X), g_{Y\in \text{Att}(X)}(\{1 - V_e(Y)\}))$ and computed by the algorithm $\Pi$. Since the computation of the equilibrium values of the nodes takes the values of the attacking nodes into account, in Cayrol and Lagasque-Schiex’s terminology, the algorithm $\Pi$ offers a procedure to perform an interaction-based valuation of the graph $(S, R)$. However, our approach is more general because the computation is done in terms of equations satisfying abstract principles.

We start our discussion with a simple graph without cycles, such as the one in Figure 13 to illustrate how numerical argumentation networks are used in the context of the argumentation-friendly functions seen in this paper.

![Figure 13: A simple argumentation graph without cycles.](image)

Given initial values $V_0(X)$, $V_0(Y)$, and $V_0(Z)$ for the nodes $X$, $Y$ and $Z$, respectively, we want the values of $V_e(X)$, $V_e(Y)$ and $V_e(Z)$ to depend on them. Since the node $X$ is not attacked by any node, its equilibrium value $V_e(X)$ is defined as $h(V_0(X), g(\emptyset)) = h(V_0(X), 1) = V_0(X)$. However, the value of $V_e(Y)$ and $V_e(Z)$ depend not only on their initial values, but also on the equilibrium values of their attackers. This suggests some notion of directionality in the computation.

Now consider a more complex network, in which the node $X$ has a number of attackers as well as an initial value $V_0(X)$ as depicted in Figure 14.

![Figure 14: Attacks to a node and its initial value.](image)

We can compute $g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}) = y$, which gives us the value of the attack on $X$. The equilibrium value of $X$ is the result of combining its initial value $V_0(X)$ with the value of the combined attacks on it, so we can pretend we have the interaction depicted in Figure 15.

64
and compute \( h(V_e(Z_1), V_e(Z_2)) \), i.e., \( h(V_0(X), g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\})) \). We get equations of the kind
\[
V_e(X) = h(V_0(X), g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}))
\] (11)
to solve. As we mentioned, \( g \) and \( h \) may be different functions, so for example we could have \( g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}) = \min(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}) \) and \( h(x, y) = x \cdot y \).

When \( f \) and \( g \) are the same, e.g., \( f = g = \min \), we can pretend we have Figure 16. And then we get \( V_e(X) = \min(\{1 - (1 - V_0(X)), 1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}) = \min(V_0(X), 1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)) \). Note that in this situation, the traditional equation (without \( h \) and initial values) is a special case of \( V_0(X) = 1 \), because \( h(1, z) = z \) and then \( V_e(X) = h(1, g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\})) = g(\{1 - V_e(Y_1), \ldots, 1 - V_e(Y_k)\}) \).

We now address another issue. Once we solve equation (11), we get a function \( V_e \) such that
\[
V_e(X) = h(V_0(X), g(\{1 - Y_1, \ldots, 1 - Y_k\}))
\]
Can we use $V_e(X)$ itself as an initial value? In other words, do we have that equation (12) below holds?

$$V_e(X) = h(V_e(X), g(\{1 - Y_1, \ldots, 1 - Y_k\}))$$  \hspace{1cm} (12)

The answer is “no”, because $g$ and $h$ are not necessarily the same function. In case it is the same function, we have

$$V_e(X) = h(V_e(X), g(\{1 - Y_1, \ldots, 1 - Y_k\}))
= g(V_e(X), g(\{1 - Y_1, \ldots, 1 - Y_k\}))
= g(V_e(X), 1 - Y_1, \ldots, 1 - Y_k)
= g(Z, 1 - Y_1, \ldots, 1 - Y_k)$$

where $Z$ is the equilibrium value of a new point attacking $X$, whose value is fixed at $V_0(X)$. We can simulate this by adding new points $Z_1^X$ and $Z_2^X$ for each $X$ and form the graph depicted in Figure 17. All solutions to the cycle

![Figure 17: Combining attacks and initial value.](image)

$Z_1^X \leftrightarrow Z_2^X$ are of the form $(V_e(Z_1^X), 1 - V_e(Z_1^X))$, which means that $Z_1^X$ can get any value in $U$ and hence so can its attack on $X$. This can be seen as having the same effect as giving $X$ a particular initial value in $U$.

These conditions are satisfied by the t-norm $\min$. An attack takes the complement of the value of the attacking node to 1 (co-norm).

We have that

$$\min_{Y \in \mathcal{A}(X)} \{1 - V_e(Y)\} = 1 - \max_{Y \in \mathcal{A}(X)} \{V_e(Y)\}$$

giving us our now familiar $E_{\max}$.

The t-norm $\min$ only cares about the strength of the strongest argument. In some applications, one could argue that attacks by multiple arguments should bear more weight than the value of any of the arguments alone. One way of modelling this is by combining attacks via $product$.

$$\prod_{Y \in \mathcal{A}(X)} (1 - V_e(Y))$$  \hspace{1cm} (13)

66
Again, if any attacker of an argument has equilibrium value 1, then the value of the product will be 0. Otherwise, if all attackers of \( X \) are fully defeated, i.e., if they all have equilibrium value 0, then the value of the product will be 1. Combining the value of attacks in this way was initially proposed in \([1]\).

The expression (13) is equivalent to

\[
1 - \forall_{Y \in \text{Att}(X)} V_e(Y)
\]

where \( x \forall y = x + y - x \cdot y \) and for \( V = \{x_1, \ldots, x_k\} \), \( \forall V = (((x_1 \forall x_2) \forall \ldots) \forall x_k) \).

(14) is the complement of the probabilistic sum t-conorm. It is well known that in probability theory, the probabilistic sum expresses the probability of the occurrence of independent events. Since we want to weaken the value of the attacked node, we take the complement of this sum to 1.

A network generates a system of equations. If there are cycles in the graph, then some of the variables associated with equilibrium values will be expressed in terms of each other. We now explore this in a bit more detail.

Consider the following example.

Assume that all initial values are 1, that \( g \) and \( h \) are product. The graph in Figure 18 will generate the system of equations

\[
\begin{align*}
V_e(X) & = 1 - V_e(Y) \\
V_e(Y) & = 1 - V_e(X)
\end{align*}
\]

which has an infinite number of solutions given by the formula \( V_e(X) + V_e(Y) = 1 \). A way to arrive at a unique solution to the equations is to introduce a constant \( \kappa < 1 \) and analyse the solution to the system of equations in the limit \( \kappa \to 1 \). This would give us
\[ Ve(X) = \kappa(1 - Ve(Y)) \]
\[ Ve(Y) = \kappa(1 - Ve(X)) \]

\[ Ve(X) = \kappa - \kappa Ve(Y) \]
\[ = \kappa - \kappa(\kappa - \kappa Ve(X)) \]
\[ = \kappa - \kappa^2 + \kappa^2 Ve(X) \]
\[ Ve(X) - \kappa^2 Ve(X) = \kappa - \kappa^2 \]
\[ Ve(X)(1 - \kappa^2) = \kappa - \kappa^2 \]
\[ Ve(X) = \frac{\kappa(1 - \kappa)}{(1 - \kappa)(1 + \kappa)} \]
\[ Ve(X) = \frac{\kappa}{1 + \kappa} \]

Hence, when \( \kappa \to 1 \), \( Ve(X) = Ve(Y) = \frac{1}{2} \). This result explains the implicit introduction of the parameter \( \varepsilon \) to the vote aggregation function proposed by Leite and Martins in [17].

Since the initial values of the two nodes in the network of Figure [18] are the same, another way of looking at the network is by unravelling the cycle starting arbitrarily at one of its nodes, say \( X \). In our example, this would result in the (infinite) network of Figure [19].

\[ ... \]

Figure 19: Unravelling the cycle in the network of Figure [18].

If we assume the initial values for \( X \) and \( Y \) are both \( x \), the equilibrium value for \( X \) could be calculated as

\[ Ve(X) = x \cdot (1 - (x \cdot (1 - (x \cdot (1 - \ldots)))))) \]

Now suppose \( x = \frac{1}{1 + \varepsilon} \), for some \( \varepsilon > 0 \), we have that

\[ Ve(X) = \frac{1}{1 + \varepsilon}(1 - \left( \frac{1}{1 + \varepsilon}(1 - \left( \frac{1}{1 + \varepsilon}(1 - \ldots)) \right) \right) \]

Thus, in fact, we would be multiplying the initial value \( x = \frac{1}{1 + \varepsilon} \) by the number

\[ \delta = 1 - \left( \frac{1}{1 + \varepsilon}(1 - \left( \frac{1}{1 + \varepsilon}(1 - \ldots)) \right) \right) \]

\[ We\ disagree\ with\ the\ reasons\ for\ the\ introduction\ of\ the\ parameter\ itself,\ although\ technically\ it\ is\ the\ reason\ why\ the\ solution\ converges.\ A\ full\ discussion\ about\ this\ is\ given\ on\ Section 5\]
Let us calculate what the value $\delta$ is. To simplify the calculation we set $\alpha = (1 + \varepsilon)$, we then get

$$\delta = 1 - \left( \frac{1}{\alpha} (1 - \frac{1}{\alpha}(1 - \ldots)) \right)$$

If we expand the first multiplication, we get

$$\delta = 1 - \left( \frac{1}{\alpha} - \frac{1}{\alpha^2} (1 - \frac{1}{\alpha}(1 - \ldots)) \right)$$

$$= 1 - \left[ \frac{1}{\alpha} - \frac{1}{\alpha^2} + \frac{1}{\alpha^3} (1 - \frac{1}{\alpha}(1 - \ldots)) \right]$$

$$= 1 - \left[ \frac{1}{\alpha} - \frac{1}{\alpha^2} + \frac{1}{\alpha^3} - \frac{1}{\alpha^4} (1 - \frac{1}{\alpha}(1 - \ldots)) \right]$$

$$= 1 - \left[ \left( \frac{\alpha - 1}{\alpha^2} \right) + \left( \frac{\alpha - 1}{\alpha^4} \right) + \left( \frac{\alpha - 1}{\alpha^6} \right) + \ldots \right]$$

The component

$$\left( \frac{\alpha - 1}{\alpha^2} \right) + \left( \frac{\alpha - 1}{\alpha^4} \right) + \left( \frac{\alpha - 1}{\alpha^6} \right) + \ldots$$

can be re-written as

$$\sum_{k=1}^{\infty} (\alpha - 1) \left( \frac{1}{\alpha^2} \right)^k$$

which is the same as

$$\sum_{k=0}^{\infty} (\alpha - 1) \left( \frac{1}{\alpha^2} \right)^k - (\alpha - 1)$$

The first component in the main subtraction above is the sum of a geometric series with common ratio $\frac{1}{\alpha^2}$ and scale factor $\alpha - 1$. Now note that the ratio $\frac{1}{\alpha^2} < 1$, since $\alpha = 1 + \varepsilon > 1$, and hence

$$\sum_{k=0}^{\infty} (\alpha - 1) \left( \frac{1}{\alpha^2} \right)^k = \frac{(\alpha - 1)}{1 - \frac{1}{\alpha^2}} = \frac{\alpha^2 (\alpha - 1)}{\alpha^2 - 1}$$

The subtraction can therefore be re-written as

$$\frac{\alpha^2 (\alpha - 1)}{\alpha^2 - 1} - (\alpha - 1)$$

$$= \frac{\alpha^2 (\alpha - 1) - (\alpha^2 - 1)(\alpha - 1)}{\alpha^2 - 1}$$

$$= \frac{(\alpha - 1)(\alpha^2 - \alpha^2 + 1)}{\alpha^2 - 1}$$

$$= \frac{\alpha}{\alpha^2 - 1}$$
Remember that $\alpha = 1 + \varepsilon$, hence
\[
\frac{\alpha}{\alpha^2 - 1} = \frac{1 + \varepsilon - 1}{(1 + \varepsilon)(1 + \varepsilon) - 1} = \frac{\varepsilon^2 + 2\varepsilon + 1 - 1}{\varepsilon(\varepsilon + 2)} = \frac{1}{\varepsilon + 2}
\]

Therefore,
\[
\delta = \left(1 - \frac{1}{\varepsilon + 2}\right)
\]
and hence in the limit $\varepsilon \to 0$, we get
\[
V_e(X) = \lim_{\varepsilon \to 0} \frac{1}{1 + \varepsilon} \left(1 - \frac{1}{\varepsilon + 2}\right) = \frac{1}{2}
\]
as expected.

If we just have an acyclic sequence of attacks such as the one in Figure 20, we can analyse what happens with the equilibrium values of each node, given a fixed initial value $v$ for all nodes (again we consider $f$ as product).

\[
\begin{array}{c}
X_1 \\
\rightarrow \\
X_2 \\
\rightarrow \\
\cdots \\
\rightarrow \\
X_k
\end{array}
\]

Figure 20: Sequence of attacks.

From the network in Figure 20, we get that $V_e(X_1) = v$, $V_e(X_2) = v \cdot (1 - v)$, $V_e(X_3) = v \cdot (1 - (v \cdot (1 - v)))$, and so forth. If $v = 1$, then $V_e(X_1) = 1$, $V_e(X_2) = 0$, $V_e(X_3) = 1$, ..., . The values alternate between 0 and 1, agreeing with Dung’s original semantics as expected. If $v = 0$, then $V_e(X_i) = 0$ for all $0 \leq i \leq k$. This is a consequence of the fact, that by using $g$, the equilibrium value depends on the node’s initial value and if this is 0, so is the equilibrium value of the node when $g$ is product. Similarly, if the initial values of all nodes is $\frac{1}{2}$, we get $V_e(X_1) = \frac{1}{2}$, $V_e(X_2) = \frac{1}{4}$, $V_e(X_3) = \frac{3}{8}$, ..., .

Contrast the calculation of the equilibrium values above with that of Besnard and Hunter [2], in which the values are calculated by a so-called categoriser function. In their paper, the given example of such a function was the $h$-categoriser $h$, defined as
\[
h(X) = \begin{cases} 
1, & \text{if } \text{Att}(X) = \emptyset \\
\frac{1}{1 + \sum_{Y \in \text{Att}(X)} h(Y)}, & \text{otherwise}
\end{cases}
\]

Assuming initial value $v = 1$ in the example above, we would have that $h(X_1) = 1$, $h(X_2) = \frac{1}{2}$, $h(X_3) = \frac{2}{3}$, and so forth. This obviously does not agree with Dung’s interpretation.
The effect on the equilibrium value of a node calculated using $g$ and $h$ as product, when the node is attacked by a single node of same initial value is now discussed. This is the scenario depicted in Figure 21.

![Figure 21: Attack by a node of same initial value.](image)

If we assume that $X$ and $Y$ get initial value $x$, we have that since $X$ has no attacking arguments, $V_e(X) = x \cdot (1 - 0) = x$. We then have:

\[
V_e(X) = x \\
V_e(Y) = x(1 - V_e(X)) = x - x^2
\]

If $X$ gets initial value 1, then it gets equilibrium value 1 and since it attacks $Y$, its equilibrium value is 0, as expected. On the other hand, if $X$ and $Y$ get initial value 0, then $Y$’s equilibrium value will also be 0. If $X$ and $Y$ get initial value $\frac{1}{2}$, then the attack by $X$ on $Y$ is not sufficiently strong to annihilate $Y$’s initial value completely. In fact, it only brings it down by 50%, i.e., giving it equilibrium value $\frac{1}{4}$. This is the maximum weakening that an attack by an equally strong argument can inflict on $Y$ using product. The full range of values under these circumstances is illustrated by Figure 22.

![Figure 22: Attack by a single node of same initial value.](image)

---

\textsuperscript{12}This equilibrium value would be 0 independently of the initial value of $Y$ in this case, because we retain Dung’s semantics in the trivial cases.
B.1 Comparisons with Social Abstract Argumentation Networks

In [17], Leite and Martins proposed social abstract argumentation frameworks which can be seen as an extension of Dung’s abstract argumentation frameworks to allow the representation of information about votes to arguments. This work was subsequently extended in [11] to handle votes on attacks too.

The motivation for these networks is to provide a means to calculate the result of the interaction between arguments using approval and disapproval ratings from users of news forums. The idea is that when a user sees an argument, she may approve it, disapprove it, or simply abstain from expressing an opinion. Since the arguments relate to each other through an attack relation (not necessarily known to the users), the votes themselves are not sufficient to provide an overall picture of the discussion. An interesting feature of these environments is therefore their intrinsic informal nature in the sense that in practice it is possible that voters vote for multiple arguments in the debate and also that users may be unaware of conflicts between the arguments.

One immediate concern is the provision of an appropriate semantics which can give an interpretation to the votes capturing the intuition of the voting process. The semantics must take into account both the interactions between the arguments as well as the votes originally cast for them.

We now introduce Eğilmez et al.’s work [11], which is an extension to [17] so we can compare it with our methodology.

Definition B.2 [11] A social abstract argumentation framework is a tuple $\langle S, R, V_S, V_R \rangle$, where $S$ is a set of arguments; $R : S \times S$ is a binary attack relation between arguments; and $V_S : S \rightarrow \mathbb{N} \times \mathbb{N}$ and $V_R : R \rightarrow \mathbb{N} \times \mathbb{N}$ are functions mapping arguments and attacks to tuples $\langle v^+, v^- \rangle$ representing the total of approval and disapproval votes received by each.

In order to provide a semantical interpretation, Eğilmez et al. introduce the concept of a semantic framework presented below.

Definition B.3 [11] A social abstract argumentation semantic framework is a tuple $\langle L, \tau, \lambda, \gamma, \neg \rangle$, where

- $L$ is a totally ordered set with top and bottom elements $\top$ and $\bot$, respectively
- $\tau : \mathbb{N} \times \mathbb{N} \rightarrow L$ is a vote aggregation function that computes the social support of arguments and attacks
- $\lambda_S, \lambda_R : L \times L \rightarrow L$; $\gamma : L \times L \rightarrow L$; and $\neg : L \rightarrow L$ are algebraic operations on $L$.

13Note that [11] were not aware (and did not quote) [1], which was six years earlier. Thus, the only new contribution in [11] was how they determine the initial values and the connection with voting.
The operations $\tau$, $\land$, $\lor$ and $\neg$ are used to calculate the overall strength of the arguments and attacks based on their initial votes. For the voting scenario considered in [11], the so-called product semantics was given. In this semantics, $L$ is $U$ (i.e., the interval $[0, 1]$); $\land_S$ and $\land_R$ are both the product t-norm $\land$, where $x \land y = x \cdot y$; $\lor$ is its associated t-conorm, i.e., $x \lor y = 1 - (1 - x) \cdot (1 - y) = x + y - x \cdot y$; $\neg x = 1 - x$; and $\tau$ is one of a family of operations $\tau_\varepsilon$ defined as follows:

**Definition B.4** [Initial support for attacks and arguments] Let $X$ be an argument and $V_S(X) = \langle p, m \rangle$.

$$\tau_\varepsilon(X) = \frac{p}{p + m + \varepsilon}$$

where $\varepsilon > 0$.

The initial support value for an attack $(X, Y)$ is calculated identically, except that we use $V_R((X, Y))$ instead of $V_S(X)$.

One can regard $\tau_\varepsilon$ and the operation that calculates the initial social support value for arguments and attacks. However, one adverse effect of calculating the initial support in this way is that it fails to put the votes in context, so an argument for which a single supporting vote is cast can get social support close to 1 (depending on what the value of $\varepsilon$ is).

The semantics of a social abstract framework is then defined by a social model presented below.

**Definition B.5** [11] Let $F$ be a social abstract argumentation framework and $\mathcal{T} = \langle L, \tau, \land_S, \land_R, \lor, \neg \rangle$ a semantic framework. A social model of $F$ under semantics $\mathcal{T}$ is a total mapping $M : S \rightarrow L$ such that for every $X \in S$

$$M(X) = \tau(X) \land \neg \land Y_i \in \text{Att}(X) \{ \tau((Y_i, X)) \land M(Y_i) \}$$

Note that if $\land$ is product t-norm and $\lor$ is its t-conorm, as in [11], then

$$M(X) = \tau(X) \land \neg \land Y_i \in \text{Att}(X) \{ \tau((Y_i, X)) \land M(Y_i) \}$$

$$= \tau(X) \cdot \left(1 - \prod_{Y_i \in \text{Att}(X)} (1 - \tau((Y_i, X)) \cdot M(Y_i))\right)$$

$$= \tau(X) \cdot \prod_{Y_i \in \text{Att}(X)} (1 - \tau((Y_i, X)) \cdot M(Y_i))$$

Contrast $M(X)$ with the equilibrium value of $X$, $V_e(X)$ as we proposed it in [11] Definition 5:

$$V_e(X) = V_0(X) \cdot \prod_{Y_i \in \text{Att}(X)} (1 - \xi((Y_i, X)) V_e(Y_i))$$

$\varepsilon$ cannot be 0, because this would render $\tau_\varepsilon$ ill defined for components with no votes.
The calculation is exactly the same, except that we compute initial support differently as discussed next. We emphasise that the notion of the strength of attack already existed since [1].

As Leite et al. initially pointed out in [17], there are difficulties with the vote aggregation function $\tau$. At first, the constant $\varepsilon$ was introduced to avoid the existence of infinite models. For example, consider the network

![Network Diagram]

And assume that $V_S(X) = V_S(Y) = \langle x, 0 \rangle$. Then we have that $\tau_0(X) = \tau_0(Y) = 1$ and hence any model $M$ satisfying the equation $M(X) = 1 - M(Y)$ is a social model of the network.

However, if the social support uses a very small value for $\varepsilon$ that is nevertheless greater than 0, we get the following situation.

\[
M(X) = \frac{1}{1 + \varepsilon} (1 - M(Y)) \\
M(Y) = \frac{1}{1 + \varepsilon} (1 - M(X))
\]

If we substitute one value for the other, we get that

\[
M(X) = \frac{1}{1 + \varepsilon} \left( 1 - \frac{1}{1 + \varepsilon} (1 - M(X)) \right) \\
= \frac{1}{1 + \varepsilon} \left( \frac{1 + \varepsilon - 1 + M(X)}{1 + \varepsilon} \right) \\
= \frac{1}{1 + \varepsilon} \left( \frac{\varepsilon + M(X)}{1 + \varepsilon} \right) \\
= \frac{\varepsilon + M(X)}{(1 + \varepsilon)^2}
\]

\[
M(X)(1 + \varepsilon)^2 = \varepsilon + M(X) \\
M(X)(1 + \varepsilon)^2 - M(X) = \varepsilon \\
M(X) = \frac{\varepsilon}{(1 + \varepsilon)^2 - 1} \\
= \frac{\varepsilon}{2\varepsilon + \varepsilon^2} \\
= \frac{1}{2 + \varepsilon}
\]
and hence $\lim_{\varepsilon \to 0} M(X) = \frac{1}{2} = M(Y)$, which provides a unique solution.

In our opinion, there is a methodological problem and a technical one. The value $\varepsilon > 0$ solves the technical problem, which is the convergence to a single model. However, methodologically speaking, the objective of $\tau$ is to calculate initial support for components and in that respect, the constant $\varepsilon$ has no part to play. This situation does not arise in [16] [15], because the social support function there is normalised with respect to the total number of argumentation networks being merged. We hope we have shed some light into the technicalities of finding solutions to the equations throughout this paper.

A more difficult problem is the exaggerated role played by terminal arguments with little support, as shown below. Consider the following example:

![Diagram](image)

and assume that $V_S(X) = \langle 1, 0 \rangle$ and $V_S(Y) = \langle 99, 0 \rangle$. According to Definition B.4, $\tau_0(X) = 1$. Since $X$ is a terminal argument, $M(X) = 1(1 - 0) = 1$ and hence $M(Y) = \tau_0(Y)(1 - \tau((X,Y)) \cdot M(X)) = \tau_0(Y)(1 - \tau((X,Y)))$. Hence, the fate of $Y$ depends on how strongly the attack from $X$ is supported.\footnote{The main motivation for the introduction of the weights on attacks in [11].} Although this technically solves the problem, it mixes the two issues, because a voter must vote for an argument as well as for its attacks, if they are to have any effect and an argument can get very high initial support even if it is voted only by a very small number of voters.$^{10}$

\footnote{High values of $\tau$ should correspond to high level of initial support.}