

Equilibrium States in Numerical Argumentation Networks

D. M. Gabbay and O. Rodrigues

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.

Mathematics Subject Classification (2010). Primary 68T27; Secondary 60B12, 68T30.

Keywords. argumentation, numerical methods, aggregation.

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 [13, 14]

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

This paper offers an iteration schema for finding specific solutions to the equations responding to initial requirements and shows what these solutions correspond to in the abstract argumentation sense.

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 area¹ 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 Newton's 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 complete extensions) 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 [7]. 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 turn 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 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 interactions and the second about merging/voting in argumentation

¹For 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.

²For 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.

³This 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 introduction on numerical analysis see [21]).

networks. We shall see that Newton’s 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

An abstract argumentation framework is a formalism proposed by Dung [10] and defined in terms of a tuple $\langle S, R \rangle$, where S is a non-empty set of arguments and $R \subseteq S \times S$ is a binary *attack relation*. We will refer to an abstract argumentation framework $\langle S, R \rangle$ simply as an *argumentation network*. If $(X, Y) \in R$, we say that the argument X attacks the argument Y . $\langle S, R \rangle$ can be seen as a directed graph (see Figure 1). As informally introduced in Section 1, $Att(X)$ will be used to denote the set $\{Y \in S \mid (Y, X) \in R\}$, i.e., the set of arguments attacking the argument X . Following graph theory convention, if X has no attackers (i.e., $Att(X) = \emptyset$), we say that X is a *source node* in $\langle S, R \rangle$. 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 $\{Y \in S \mid E \rightarrow Y\}$.

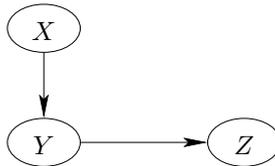


FIGURE 1. A sample argumentation network.

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 explained next.

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 complete 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 $(Y, X) \in R$.

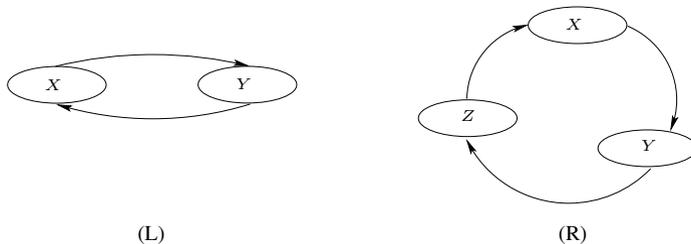


FIGURE 2. Sample argumentation networks.

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* [13, 14]. These are explained next.

The labelling semantics.

The labelling semantics uses labelling functions $\lambda : S \rightarrow \{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$ 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 $\langle S, R \rangle$ be an argumentation network and λ a labelling function for S .*

1. *An argument $X \in S$ is illegally labelled **in** by λ if $\lambda(X) = \mathbf{in}$ and there exists $Y \in \text{Att}(X)$ such that $\lambda(Y) \neq \mathbf{out}$.*
2. *An argument $X \in S$ is illegally labelled **out** by λ if $\lambda(X) = \mathbf{out}$ and there is no $Y \in \text{Att}(X)$ such that $\lambda(Y) = \mathbf{in}$.*
3. *An argument $X \in S$ is illegally labelled **und** by λ if $\lambda(X) = \mathbf{und}$ and either for all $Y \in \text{Att}(X)$, $\lambda(Y) = \mathbf{out}$ or there exists $Y \in \text{Att}(X)$, such that $\lambda(Y) = \mathbf{in}$.*

A legal (complete) labelling is a labelling in which no argument is illegally labelled.

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 λ_1 , λ_2 and λ_0 below.

$\lambda_1 \Leftrightarrow E_1 = \{X\}$	$\lambda_2 \Leftrightarrow E_2 = \{Y\}$	$\lambda_0 \Leftrightarrow E_0 = \emptyset$
$\lambda_1(X) = \mathbf{in}$	$\lambda_2(X) = \mathbf{out}$	$\lambda_0(X) = \mathbf{und}$
$\lambda_1(Y) = \mathbf{out}$	$\lambda_2(Y) = \mathbf{in}$	$\lambda_0(Y) = \mathbf{und}$

For the network (R), we have only the function λ such that $\lambda(X) = \lambda(Y) = \lambda(Z) = \mathbf{und}$. 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 $U = [0, 1]$. 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 $E_{q_{\max}}$ below, where $V(X)$ is the value of a node $X \in S$:

$$(E_{q_{\max}}) \quad V(X) = 1 - \max_{Y_i \in Att(X)} \{V(Y_i)\}$$

Another possibility is $E_{q_{\text{inv}}}$:

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

It is easy to see that according to $E_{q_{\max}}$ 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

$$\begin{aligned} V(X) &= 1 - V(Y) \\ V(Y) &= 1 - V(X) \end{aligned}$$

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.

In fact, Gabbay has shown that in the case of $E_{q_{\max}}$ the totality of solutions to the system of equations corresponds to the totality of extensions in Dung's sense [14]. The correspondence is best explained in terms of the labelling semantics, using the following correspondence:

$$\begin{aligned} V(X) = 1 & \quad :: \quad \lambda(X) = \mathbf{in} \\ V(X) = 0 & \quad :: \quad \lambda(X) = \mathbf{out} \\ 0 < V(X) < 1 & \quad :: \quad \lambda(X) = \mathbf{und} \end{aligned}$$

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_{\max} and Eq_{inv} embed 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 $\text{Att}(X) = \{Y_1, Y_2, \dots, Y_k\}$. 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*.⁴ 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.

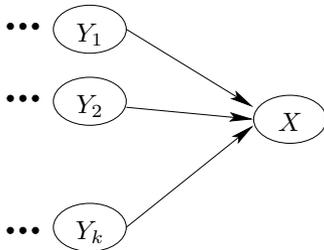


FIGURE 3. Multiple attacks on a node.

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 Eq_{\max} . Note that the conjunction operation in the schema Eq_{inv} is *product*. The operations min and product are two

⁴If we make **und** equals $\frac{1}{2}$, then an attack by a single undecided node will have value $\frac{1}{2}$.

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 with domain being the family of all finite sequences of elements from U and range U is argumentation-friendly if g satisfies the following conditions.*

- (T1) $g(\emptyset) = 1$
- (T2) $g(1; \Delta) = g(\Delta)$ ⁵
- (T3) $g(\langle x_1, \dots, x, \dots, y, \dots, x_n \rangle) = g(\langle x_1, \dots, y, \dots, x, \dots, x_n \rangle)$
- (T4) $g(\Delta) = 0$ if and only if $0 \in \Delta$
- (T5) $g(\Delta) = 1$ if and only if $x = 1$ for every $x \in \Delta$
- (T6) g is continuous as a multi-variable function⁶

Example 1.3. *Below are some examples of argumentation-friendly functions:*

1. $g(\Delta) = \begin{cases} 1, & \text{if } \Delta = \emptyset \\ \min\{x_i\}, & \text{if } \Delta = \langle x_1, \dots, x_n \rangle \end{cases}$
2. $g(\Delta) = \begin{cases} 1, & \text{if } \Delta = \emptyset \\ \prod_1^n (1 - x_i), & \text{if } \Delta = \langle x_1, \dots, x_n \rangle \end{cases}$
3. $g_\lambda(\Delta) = (1 - \lambda) \min\{\frac{1}{2}, g(\Delta)\} + \lambda \max\{\frac{1}{2}, g(\Delta)\}$, for any g satisfying (T1)–(T6).

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

Theorem 1.5. *Let $\mathcal{N} = \langle S, R \rangle$ be a network, g an argumentation-friendly function, and \mathbf{T} a system of equations written for \mathcal{N} , where for each node X , $V(X) = g_{Y \in Att(X)}(\{1 - V(Y)\})$. Take any solution V to \mathbf{T} , it follows that $in(V)$ is a complete extension.*

Proof. Suppose that $in(V)$ is not conflict-free. Then there are $X, Y \in in(V)$, such that $(X, Y) \in R$. Since $Y \in in(V)$, then $V(Y) = 1 = g_{W \in Att(Y)}(\{1 - V(W)\})$. But $X \in Att(Y)$ and $X \in in(V)$, and hence $V(X) = 1$. It then follows by (T4) that $g(\langle \dots, 0, \dots \rangle) = 0$ and hence $1 \neq 0$, a contradiction.

Now suppose that $X \in in(V)$. We show that for all $Y \in Att(X)$ there exists $Z \in in(V)$, such that $(Z, Y) \in R$. If $V(X) = 1$, then $g_{Y \in Att(X)}(\{1 -$

⁵The values of g for any sequence containing the value 1 is the same as the value of g for the subsequence without the 1.

⁶In fact, this condition is only needed to guarantee the existence of solutions to the equations.

$V(Y)\} = 1$ and then by (T5) it follows that $1 - V(Y) = 1$, for all $Y \in \text{Att}(X)$ and hence $V(Y) = 0$ for all $Y \in \text{Att}(X)$. Take any such Y . Since $V(Y) = 0$, we have by (T4) that for some $W \in \text{Att}(Y)$, $V(W) = 1$. It then follows that $W \in \text{in}(V)$. \square

Theorem 1.6. *Let $\mathcal{N} = \langle S, R \rangle$ be a network, g an argumentation-friendly function, and \mathbf{T} a system of equations written for \mathcal{N} , where for each node X , $V(X) = g_{Y \in \text{Att}(X)}\{1 - V(Y)\}$. Then for every preferred extension $E_{\mathcal{N}}$ of \mathcal{N} , there exists a solution V to \mathbf{T} such that*

- (C1) *If $X \in E_{\mathcal{N}}$, then $V(X) = 1$*
- (C2) *If $E_{\mathcal{N}} \rightarrow X$, then $V(X) = 0$*
- (C3) *If $X \notin E_{\mathcal{N}}$ and $E_{\mathcal{N}} \not\rightarrow X$, then $0 < V(X) < 1$*

Proof. Let us start by partitioning the set S using $E_{\mathcal{N}}$ into three sets $\Delta_1 = E_{\mathcal{N}}$, $\Delta_0 = \{X \in S \mid E_{\mathcal{N}} \rightarrow X\}$, and $\Delta_u = S \setminus (\Delta_0 \cup \Delta_1)$. Note that the elements of Δ_u are the undecided elements in S with respect to $E_{\mathcal{N}}$. Each element of Δ_u is not attacked by any element of Δ_1 and its attackers cannot all come from Δ_0 , i.e., at least one attacker comes from Δ_u itself. Consider the argumentation network $\langle \Delta_u, R \upharpoonright \Delta_u \rangle$. Write a system of equations \mathbf{T}_u using g for $\langle \Delta_u, R \upharpoonright \Delta_u \rangle$. For each $X \in \Delta_u$, the equation is

$$V_u(X) = g_{Y \in \Delta_u \text{ s.t. } (Y, X) \in R \upharpoonright \Delta_u} \{1 - V_u(Y)\}$$

By Brouwer's theorem, the above equations have a solution V_u .⁷ To be clear V_u is defined on Δ_u , giving values $V_u(X)$, such that for every $X \in \Delta_u$, $V_u(X) = g_{Y \in \Delta_u \text{ s.t. } (Y, X) \in R \upharpoonright \Delta_u} \{1 - V_u(Y)\}$

We are seeking however a solution V defined for all of $S = \Delta_0 \cup \Delta_1 \cup \Delta_u$, which satisfies the system of equations \mathbf{T} for $\langle S, R \rangle$:

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

Furthermore, we want V to be such that $V(X) = 1$ for $X \in \Delta_1$, $V(X) = 0$, for $X \in \Delta_0$ and $V(X) \in (0, 1)$ for $X \in \Delta_u$. We now define such a solution V . Let

$$\begin{aligned} V(X) &= 1, & \text{for all } X \in \Delta_1 \\ V(X) &= 0, & \text{for all } X \in \Delta_0 \\ V(X) &= V_u(X), & \text{for all } X \in \Delta_u \end{aligned}$$

We have to show now that V indeed solves the system of equations \mathbf{T} for $\langle S, R \rangle$. Take $X \in S$:

Case 1: $X \in \Delta_1$. We defined $V(X) = 1$. We need to show that $1 = g_{Y \in \text{Att}(X)} \{1 - V(Y)\}$. Since $X \in E_{\mathcal{N}}$, then all of its attackers are in Δ_0 , and then $V(Y) = 0$ (by definition), for all $Y \in \text{Att}(X)$. Therefore, $g_{Y \in \text{Att}(X)} \{1 - V(Y)\} = 1$, by (T5).

⁷The Euclidean version of the theorem states that if g is a real-valued function, defined and continuous on a bounded closed interval I of the real line where $g(x) \in I$, for all $x \in I$, then g has a fixed-point. In our case, there are $n = |S|$ variables in the network $\langle S, R \rangle$, which we can associate with the vector \vec{X} . We can then see each equation as $\vec{X} = \vec{g}(\vec{X})$, where \vec{g} is a continuous function on the n -dimensional space $[0, 1]^n$ (see Theorem 1.2 in [21]).

Case 2: $X \in \Delta_0$. We defined $V(X) = 0$. We need to show that $0 = g_{Y \in Att(X)}\{1 - V(Y)\}$. Since $E_{\mathcal{N}} \rightarrow X$, then there exists $Y \in Att(X)$, such that $Y \in \Delta_1$. By definition, $V(Y) = 1$, and then $g_{Y \in Att(X)}\{1 - V(Y)\} = 0$, by (T4).

case 3: $X \in \Delta_u$. We defined $V(X) = V_u(X) = g_{Y \in \Delta_u \text{ s.t. } (Y,X) \in R \upharpoonright \Delta_u}\{1 - V_u(Y)\}$. We need to show that $g_{Y \in Att(X)}\{1 - V(Y)\} = g_{Y \in \Delta_u \text{ s.t. } (Y,X) \in R \upharpoonright \Delta_u}\{1 - V_u(Y)\}$. We noted above, that $X \in \Delta_u$ implies that none of its attackers belong to Δ_1 and therefore any remaining attackers Z not in Δ_u must be in Δ_0 . By definition, $V(Z) = 0$, therefore $1 - 0 = 1$ and by (T2), such values can be safely deleted in the calculation of $g_{Y \in Att(X)}\{1 - V(Y)\}$. Therefore, deleting all such values will show that $g_{Y \in \Delta_u \text{ s.t. } (Y,X) \in R \upharpoonright \Delta_u}\{1 - V_u(Y)\} = g_{Y \in \Delta_u \cup \Delta_0 \text{ s.t. } (Y,X) \in R}\{1 - V_u(Y)\}$.

Having shown that V above solves the system of equations \mathbf{T} , we can use Theorem 1.5 to show that $in(V)$ is a complete extension. We now ask whether any of the values $V_u(X)$, for $X \in \Delta_u$ can be 0 or 1. The answer is no, for if $V_u(X) = 1$ for any $X \in \Delta_u$, then $V(X) = 1$ and then $X \in in(V) \setminus E_{\mathcal{N}}$, which is impossible, since $E_{\mathcal{N}}$ is a preferred extension. Analogously, we can only get $V(X) = 0$ for some $X \in \Delta_u$, if for some of its attackers $Z \in \Delta_u$, $V(Z) = 1$, which as we mentioned is impossible. This completed the proof. \square

The condition of preferred extension of the Theorem 1.6 is necessary, as shown in the example below.

Example 1.7. Consider the complete extension $E = \{X\}$ of the network below. E is not preferred, since E is a proper subset of $\{X, W\}$.



The network generates the following equations.

$$V(X) = 1 - V(Y) \quad (1)$$

$$V(Y) = 1 - V(X) \quad (2)$$

$$V(W) = 1 - V(Z) \quad (3)$$

$$V(Z) = g(\{1 - V(W), 1 - V(Z)\}) \quad (4)$$

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 preferred extensions $\{X, W\}$ and $\{Y, W\}$ include W . No extension can include Z .

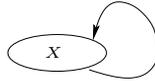
However, with g as min, we have that (4) becomes

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

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 1.8. Consider the network with a single self-referencing loop below.



The network generates the equation:

$$V(X) = g(\{1 - V(X)\})$$

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

This follows from Theorem 1.5. What it means is that any solution to the system of equations defined in terms of Eq_{\max} 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 1.10. Let $\mathcal{N} = \langle S, R \rangle$ be a network and \mathbf{T} a system of equations written for \mathcal{N} , where for each node X , $V(X) = \min_{Y \in \text{Att}(X)}(\{1 - Y\})$. Then for every complete extension E of \mathcal{N} , there exists a solution V to \mathbf{T} satisfying:

- (C1) If $X \in E$, then $V(X) = 1$.
- (C2) If $E \rightarrow X$, then $V(X) = 0$.
- (C3) If $X \notin E$ and $E \not\rightarrow X$, then $0 < V(X) < 1$.

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 \mathbf{T} . As in Theorem 1.6, replacing the above values in the original system of equations will reduce them to the following types.

- (1) $1 = \min(\Delta_1)$
- (2) $0 = \min(\Delta_2)$
- (3) $\frac{1}{2} = \min(\Delta_3)$

We have seen that $\Delta_1 = \{1\}$ and since $1 = \min(\{1\})$, (1) is satisfied. Similarly, $0 \in \Delta_2$ and since $\min(\{0, \dots\}) = 0$, so is (2). Notice that the image of V is $\{0, \frac{1}{2}, 1\}$. All values in Δ_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. \square

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?⁸ 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-Rodriguez 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. These values may come attached to the nodes for different reasons. For instance, the arguments themselves may be expressed as some proof in a fuzzy logic and the initial values can represent the values of the conclusions of the proofs, or they can be obtained as the result of the merging of some networks, or they may come from some voting mechanism, etc. Whatever the reason, the initial values may or may not correspond to a complete extension in Dung’s sense and we seek a mechanism that would allow us to find the “best” possible extension corresponding to them.

Consider the equation Eq_{\max} :

$$(Eq_{\max}) \quad V(X) = 1 - \max_{Y_i \in Att(X)} \{V(Y_i)\}$$

Eq_{\max} is satisfied when the value of the node X is *legal* (in Caminada and Pigozzi’s terminology [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 of one X ’s attackers has value 1; or if the value of $X \in (0, 1)$ and at least one of X ’s attackers has value in $(0, 1)$ and no attacker of X 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**, **out** and **und** node values intact, changing illegal **in** or **out** values into **und**.⁹ To make a distinction between these classes of values, we will call the values in $\{0, 1\}$ *crisp* and the values in $(0, 1)$ *undecided*.

Now consider the following averaging function:

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

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

⁸As can be done to find the square root of numbers using Newton’s method.

⁹We will come to the correction of illegal **und** nodes later.

- (L1) 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:

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

- (L2) 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:

$$\begin{aligned}
 1 \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\} + X \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\} &= \\
 1 \cdot \min \left\{ \frac{1}{2}, 0 \right\} + 0 \cdot \max \left\{ \frac{1}{2}, 0 \right\} &= \\
 &= 0
 \end{aligned}$$

- (L3) 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 $\alpha_1 = \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\}$ and $\alpha_2 = \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\}$. 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:

$$\begin{aligned}
 (1 - X) \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\} + X \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} Y \right\} &= \\
 (1 - X) \cdot \alpha_1 + X \cdot \alpha_2 &= \\
 \alpha_1 - X \cdot \alpha_1 + X \cdot \alpha_2 &= \\
 \alpha_1 - X \cdot (\alpha_1 - \alpha_2) &= \kappa
 \end{aligned}$$

Notice that $\alpha_1 \leq \frac{1}{2}$ and $\alpha_2 \geq \frac{1}{2}$, therefore $\alpha_2 \not< \alpha_1$. If $\alpha_1 = \alpha_2$, then $\kappa = \alpha_1$ and hence $0 < \kappa \leq \frac{1}{2}$. If $\alpha_1 < \alpha_2$, then $0 < \alpha_1 < \frac{1}{2}$ and $\alpha_2 = \frac{1}{2}$. Therefore, $-\frac{1}{2} < (\alpha_1 - \alpha_2) < 0$. It then follows that $0 < \alpha_1 \leq \kappa < \frac{1}{2}$ and therefore the value of X remains in $(0, 1)$.

What (L1)–(L3) above give us is that legal labellings are preserved.¹⁰ Later on, we shall see that our iteration schema also eventually *corrects* all illegal values. It does so in two stages. In the first stage, all illegal crisp values are turned into undecided (this is done in $t \leq |S|$ iterations). In the second stage, all remaining illegal undecided values converge to whatever legal crisp values they should be, so that in the limit, all of the values in the sequence are legal. Therefore, the *Gabbay-Rodrigues Iteration Schema* introduced below

¹⁰Legal undecided values may change, although they remain in the undecided range (by (L3)).

provides a numerical iterative method to turn any initial illegal assignment of values to arguments into its closest legal assignment.¹¹

Definition 2.1. Let $\mathcal{N} = \langle S, R \rangle$ 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:

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

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.,

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

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**.

¹¹The precise definition of “closest” will be made clear in Theorem 2.37.

Therefore, the Gabbay-Rodrigues schema remains faithful to the spirit of Dung’s semantics captured through the legal Caminada labellings just as Eq_{\max} does. Its advantage over Eq_{\max} is that it can be used iteratively as we will show in the rest of this section. ¹²

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 2.2. *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

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

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

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

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 2.3. *Let $\mathcal{N} = \langle S, R \rangle$ be a network and \mathbf{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 $E \rightarrow X$, then $V_0(X) = 0$
- (C3) If $X \notin E$ and $E \not\rightarrow X$, then $V_0(X) = \frac{1}{2}$

Proof. Let E be a complete extension and suppose $V_0(X) = 1$. Then $X \in E$ and hence, *i*) either $\text{Att}(X) = \emptyset$, or *ii*) for all $Y \in \text{Att}(X)$, $E \rightarrow Y$ (since E

¹²As an *equation*, we can regard the expression (GR) just as another type of g , a special eq_{GR} .

is admissible). As a result, $1 - \max_{Y \in \text{Att}(X)} \{V(Y)\} = 1$, and hence we have that

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

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

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

Finally, if $V_0(X) = \frac{1}{2}$, then $X \notin E$ and $E \not\rightarrow X$. We must have that for all $Y \in \text{Att}(X)$, $V_0(Y) < 1$ (otherwise, we would have that $E \rightarrow X$). 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

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

□

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 2.4. *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network, \mathbf{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, \dots be a sequence of value assignments where each $V_i, i > 0$, is generated by \mathbf{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 $V_{k+1}(X) = \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y) \right\} \leq \frac{1}{2}$.
2. Suppose $V_k(X) = 1$, then $V_{k+1}(X) = \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y) \right\} \geq \frac{1}{2}$.
3. Suppose $0 < V_k(X) < 1$. We first show that $V_{k+1}(X) > 0$. Note that $0 < (1 - V_k(X)) < 1$. Therefore, we have that

$$\begin{aligned} V_{k+1}(X) &= (1 - V_k(X)) \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y) \right\} + \\ &\quad V_k(X) \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} V_i(Y) \right\} \end{aligned}$$

It is easy to see that the first component of the above sum is greater than or equal to 0, whereas the second is strictly greater than 0, and hence $V_{k+1}(X) > 0$.

Since we start with values in U , Proposition 2.2, 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

$$\begin{aligned} 0 < \alpha < 1 \\ 0 < (1 - \alpha) < 1 \\ 0 \leq x \leq \frac{1}{2} \\ \frac{1}{2} \leq y \leq 1 \end{aligned}$$

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

$$\begin{aligned} (1 - \alpha) \cdot x + \alpha \cdot y &= 1 \\ x - \alpha \cdot x + \alpha \cdot y &= 1 \\ \alpha(y - x) &= (1 - x) \\ \alpha &= \frac{1 - x}{y - x} \end{aligned}$$

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

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

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 2.6. *Let $\mathcal{N} = \langle S, R \rangle$ be a network, \mathbf{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, \dots be a sequence of value assignments where each $V_i, i > 0$, is generated by \mathbf{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

$$\begin{aligned} V_{i+1}(X) &= (1 - V_i(X)) \cdot \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_i(Y)\} \right\} + \\ &\quad V_i(X) \cdot \max \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_i(Y)\} \right\} \\ 0 &= \min \left\{ \frac{1}{2}, 1 - \max_{Y \in \text{Att}(X)} \{V_i(Y)\} \right\} \end{aligned}$$

So, $\max_{Y \in \text{Att}(X)} \{V_i(Y)\} = 1$ 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

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

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, $\max_{Y \in \text{Att}(X)} \{V_{i+1}(Y)\} = 0$, 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 2.7. 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 . A sequence of assignments $V_i : S \mapsto U$ where each $i > 0$ is generated by the Gabbay-Rodrigues Iteration Schema for \mathcal{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 $V_{k+1}(X) = V_k(X)$; and
2. k is the smallest value for which the condition above holds.

Note that if $V_k(X) \in (0, 1)$, then $V_{k+1}(X) \in (0, 1)$, for all $k \geq 0$, by Theorem 2.4.

Corollary 2.8. Consider a sequence of value assignments V_0, V_1, V_2, \dots as described in Theorem 2.6. 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 2.7 follows from Theorem 2.4 and the second condition follows from Theorem 2.6. \square

Corollary 2.9. *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network, $V_0 : S \mapsto U$ an assignment of initial values to the nodes in S and \mathbf{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 2.4. 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 2.4 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 $|S|$. \square

Corollary 2.9 shows that for some value $0 \leq k \leq |S|$, the sequence of value assignments $V_0(X), V_1(X), V_2(X), \dots$ 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 2.10. *Given an argumentation-friendly function g , we can define the Gabbay-Rodrigues Iteration Schema for g , denoted by $GR(g)$, as follows.*

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

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 GR by $GR(g)$.

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. Our problem is to find the function f' , closest to f , which also corresponds to an extension of $\langle S, R \rangle$ (for example, solves the equations generated from $\langle S, R \rangle$). 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 2.11. *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 $h : S \mapsto U$ such that the sets $in(h) \subseteq in(f)$ and $out(h) \subseteq out(f)$ are maximal and for every node $X \in S$:*

$$\text{If } h(X) = 1, \text{ then } \max_{Y \in Att(X)} \{h(Y)\} = 0; \text{ and} \quad (5)$$

$$\text{If } h(X) = 0, \text{ then } \max_{Y \in Att(X)} \{h(Y)\} = 1. \quad (6)$$

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

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 $f(X) = 1$

and

$$\text{If } g_1(X) = 1, \text{ then } \max_{Y \in Att(X)} \{g_1(Y)\} = 0; \text{ and} \quad (7)$$

$$\text{If } g_2(X) = 1, \text{ then } \max_{Y \in Att(X)} \{g_2(Y)\} = 0; \text{ and} \quad (8)$$

$$\text{If } g_1(X) = 0, \text{ then } \max_{Y \in Att(X)} \{g_1(Y)\} = 1; \text{ and} \quad (9)$$

$$\text{If } g_2(X) = 0, \text{ then } \max_{Y \in Att(X)} \{g_2(Y)\} = 1 \quad (10)$$

It follows that $in(g_1) \subseteq in(f)$ and $out(g_1) \subseteq out(f)$; and $in(g_2) \subseteq in(f)$ and $out(g_2) \subseteq out(f)$.

Let us construct an assignment $h : S \mapsto U$, such that for all $X \in S$:

$$h(X) = 1 \text{ iff } \max(g_1(X), g_2(X)) = 1 \quad (11)$$

$$h(X) = 0 \text{ iff } \min(g_1(X), g_2(X)) = 0 \quad (12)$$

$$h(X) = \frac{1}{2} \text{ iff } 0 < g_1(X) < 1 \text{ and } 0 < g_2(X) < 1 \quad (13)$$

We now show that the assignment h is a well-defined function and that $in(h) \subseteq in(f)$ and that $out(h) \subseteq out(f)$. It is easy to see that every node X gets at least one value $h(X)$. We need to show that for every node X , this value is unique and that the above inclusions are satisfied. From (13), it is easy to see that $h(X)$ is equal to $\frac{1}{2}$ if and only if both $g_1(X) \in (0, 1)$ and $g_2(X) \in (0, 1)$. To show inclusion, suppose $X \in in(h)$. Then $h(X) = 1$ and hence $\max(g_1(X), g_2(X)) = 1$. Either $g_1(X) = 1$ or $g_2(X) = 1$ (or both), and hence $f(X) = 1$. Therefore $X \in in(f)$. To show that $h(X)$ is unique in this case, it is sufficient to show that $\min(g_1(X), g_2(X)) > 0$. Suppose $\min(g_1(X), g_2(X)) = 0$, 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 $X \in out(h)$, then $h(X) = 0$ and hence $\min(g_1(X), g_2(X)) = 0$. Then either $g_1(X) = 0$ or $g_2(X) = 0$ (or both), and hence $f(X) = 0$. Therefore, $X \in out(f)$. To show that $h(X)$ is also unique in this case, it suffices to show

that $\max(g_1(X), g_2(X)) < 1$. Suppose that $\max(g_1(X), g_2(X)) = 1$, then either $g_1(X) = 1$ or $g_2(X) = 1$, in which case $f(X) = 1$, a contradiction, since f is a function.

We now show that h satisfies (5) and (6).

Suppose $h(X) = 1$. By construction, $\max(g_1(X), g_2(X)) = 1$. It follows that *i*) either $X \in in(g_1)$, and then by (7), $\max_{Y \in Att(X)} \{g_1(Y)\} = 0$. This means that for every $Y \in Att(X)$, $g_1(Y) = 0$. By (12), for every $Y \in Att(X)$, $h(Y) = 0$, and hence $\max_{Y \in Att(X)} \{h(Y)\} = 0$; or *ii*) $X \in in(g_2)$, and then by (8), $\max_{Y \in Att(X)} \{g_2(Y)\} = 0$. By (12), for $in(g_2)$ is also admissible, $Y \in out(g_2)$, and hence for every $Y \in Att(X)$, $h(Y) = 0$, and hence again $\max_{Y \in Att(X)} \{h(Y)\} = 0$. This shows that h satisfies (5).

As for (6), suppose $h(X) = 0$, then by the construction of h either $g_1(X) = 0$ or $g_2(X) = 0$ (or both). The two cases are identical. We consider only the case $g_1(X) = 0$. By (9), $\max_{Y \in Att(X)} \{g_1(Y)\} = 1$, and hence for some $Y \in Att(X)$, $g_1(Y) = 1$. By (11), we have that $h(Y) = 1$ and then $\max_{Y \in Att(X)} \{h(Y)\} = 1$.

Note that $in(g_1) \subseteq in(h)$, $out(g_1) \subseteq out(h)$, $in(g_2) \subseteq in(h)$ and $out(g_2) \subseteq out(h)$. Therefore, since every g_1 and g_2 satisfying (7)–(10) give rise to a function h as described, and the number of all such functions is finite, then there exists one such h that the sets $in(h)$ and $out(h)$ are maximal. \square

Corollary 2.12. *Let $\langle S, R \rangle$ be a network and $f : S \mapsto U$ an assignment of values to the nodes in S and $h : S \mapsto U$ the assignment such that the sets $in(h) \subseteq in(f)$ and $out(h) \subseteq out(f)$ are maximal and for every node $X \in S$:*

$$\text{If } h(X) = 1, \text{ then } \max_{Y \in Att(X)} \{h(Y)\} = 0; \text{ and} \quad (14)$$

$$\text{If } h(X) = 0, \text{ then } \max_{Y \in Att(X)} \{h(Y)\} = 1. \quad (15)$$

as given by Theorem 2.11. Then the set $in(h)$ is the largest admissible subset of $in(f)$ such that also $out(h) \subseteq out(f)$.

Proof. $in(h)$ is conflict-free: if you take $X \in in(h)$, then $h(X) = 1$ and then $\max_{Y \in Att(X)} \{h(Y)\} = 0$. Therefore, either $Att(X) = \emptyset$; or for all $Y \in Att(X)$, $h(Y) = 0$, and hence $Y \notin in(h)$.

To show that $in(h)$ is admissible, we just need to show that if $X \in in(h)$ and $Y \in Att(X)$, then there exists $Z \in Att(Y)$, such that $Z \in in(h)$. Assume that $X \in in(h)$ and $Y \in Att(X)$. By definition, $h(X) = 1$, and then by (14), $\max_{W_x \in Att(X)} \{h(W_x)\} = 0$, and hence $h(Y) = 0$. By (15), $\max_{W_y \in Att(Y)} \{h(W_y)\} = 1$. Therefore, there exists $Z \in Att(Y)$, such that $Z \in in(h)$.

The fact that $in(h)$ is the largest subset of $in(f)$ subject to $out(h) \subseteq out(f)$ comes directly from Theorem 2.11. \square

Remark 2.13. *Consider the following network.*



There is no largest admissible subset of $E = \{X, Y\}$! There are two maximal admissible subsets $E_1 = \{X\}$ and $E_2 = \{Y\}$, so the requirement that “no new **out** nodes are generated” is very important in Theorem 2.11. In terms of assignments (or labellings for that matter) this was expressed as: $out(h) \subseteq out(f)$.¹³

If we are given an assignment $f(A) = 1$ and $f(B) = 1$, there is a class of assignments h such that the sets $in(h) \subseteq in(f)$ **and** $out(h) \subseteq out(f)$ are the largest. For instance, $h(A) = h(B) = \frac{1}{2}$. In the example above, it is sufficient to set $0 < h(A) < 1$ and $0 < h(B) < 1$ (we chose the value $\frac{1}{2}$ in Theorem 2.11 simply because we wanted to show that one existed and because as we shall see the legal undecided values will end up converging to $\frac{1}{2}$).

Note, in particular that the assignment f does not satisfy the conditions of Theorem 2.11 (which guarantee by Corollary 2.12 that $in(f)$ is an admissible set). We could turn f into an admissible assignment by just flipping one of the values of A or B to 0. However, if we did this, for instance, by generating the assignment $f'(A) = 1$ and $f'(B) = 0$, then although $in(f')$ is admissible and $in(f') \subseteq in(f)$, we would not have that $out(f') = \{B\} \subseteq out(f) = \emptyset$!

This is as it should be, because an initial assignment f encodes not only which nodes we would like to be **in**, but also those that we would like to be **out**, and we cannot decide without further information to optimise on the **in**'s in detriment of the **out**'s.

Theorem 2.14. Let $\mathcal{N} = \langle S, R \rangle$ be a network and \mathbf{T} its GR system of equations. If the sequence of values V_0, V_1, \dots becomes stable at iteration k , then $in(V_k)$ is the largest admissible set such that $in(V_k) \subseteq in(V_0)$ and $out(V_k) \subseteq out(V_0)$.

Proof. We first show that $in(V_k)$ is an admissible set.

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

Now we need to show that $in(V_k)$ is indeed the maximal admissible set such that $in(V_k) \subseteq in(V_0)$ and $out(V_k) \subseteq out(V_0)$. By Theorem 2.11, there are unique maximal sets $in(V_{max}) \subseteq in(V_0)$ and $out(V_{max}) \subseteq out(V_0)$ such that

¹³If we are given just E , we may want to think of an assignment f such that $in(f) = E$ and $out(f) = \{X \mid E \rightarrow X\}$, leaving the nodes in $S \setminus (in(E) \cup out(E))$ with a value in $(0, 1)$.

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

The second case is analogous. Take $Y \in out(V_{max})$ such that $Y \notin out(V_j)$. Since $Y \in out(V_{max})$, then $Y \in out(V_{j-1})$ and hence $V_{j-1}(Y) = 0$. Since $Y \notin out(V_j)$, then $V_j(Y) > 0$. It follows that $V_j(Y) = \min\{\frac{1}{2}, 1 - \max_{Z \in Att(Y)} V_{j-1}(Z)\} > 0$. Therefore, for all $Z \in Att(Y)$ we have that $V_{j-1}(Z) < 1$ and hence there is no $Z \in Att(Y)$, such that $Z \in in(V_{j-1})$. Since $Y \in out(V_{max})$, there must be some $Z' \in Att(Y)$, such that $Z' \in in(V_{max})$, but this is a contradiction since $Z' \notin in(V_{j-1})$ and j was the first index such that there was some $X \in in(V_{max})$, such that $X \notin in(V_j)$. \square

Remark 2.15. *Given an argumentation network $\mathcal{N} = \langle S, R \rangle$, an argumentation-friendly function g , a system of equations \mathbf{T} written for \mathcal{N} using g , and an assignment $v : S \mapsto U$, which represents initial desired values, then if v corresponds to a complete extension then the above theorems tell us that the sequence of equations $V_0 = v, V_1, V_2, \dots$ will become stable at some iteration k and $V_k = v$. Otherwise, V_k is the function giving the maximal possible crisp part $in(V_k)$ and $out(V_k)$ agreeing with v such that the set $in(V_k)$ is admissible. We now have the option of extending $in(V_k)$ into a complete extension E_{comp} that is the closest extension agreeing with $in(v)$. If this extension is also preferred, then it would correspond to an assignment f' , which solves the original system of equations \mathbf{T} (by Theorem 1.6). If the extension is not preferred, then whether such an f' exists depends on the nature of the function g . Some such functions, such as \min can always find an f' for every complete extension. Others, such as product , can not always find them.¹⁴*

We will see that with the Gabbay-Rodrigues Iteration Schema, 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 $\{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$.

Definition 2.16 (Caminada-Pigozzi/Gabbay-Rodrigues Translation). *A labelling function λ and a valuation function V can be inter-defined according to the table below.*

¹⁴Product is given in Item 2. of Example 1.3. For the network $S = \{A, B\}$, $R = \{(A, B), (B, A), (B, B)\}$ and the complete extension “all undecided”, there is no solution using product .

$\lambda(X)$	\rightarrow	$V_\lambda(X)$	$V(X)$	\rightarrow	$\lambda_V(X)$
in	\rightarrow	1	1	\rightarrow	in
out	\rightarrow	0	0	\rightarrow	out
und	\rightarrow	$\frac{1}{2}$	(0, 1)	\rightarrow	und

The choice of the value $\frac{1}{2}$ 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 $\frac{1}{2}$ in the limit, and so $\frac{1}{2}$ is the natural choice.

Definition 2.17. A legal assignment V is an assignment of values $V : S \mapsto U$ such that the corresponding labelling function λ_V defined according to Definition 2.16 is also legal.

Proposition 2.18. Let λ be a labelling function and V_λ its corresponding Caminada-Pigozzi translation. If the Gabbay-Rodrigues Iteration Schema is employed using V_λ as V_0 , then for some value $k \geq 0$, the sequence of values V_0, V_1, \dots will become stable and the sets $in(V_k)$ and $out(V_k)$ will correspond to the down-admissible labelling of λ .

Proof. This follows directly from Theorem 2.14 and Corollary 2.9. \square

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 **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 $in(V_k)$ and $out(V_k)$ correspond to the largest admissible labelling such that $in(V_k) \subseteq in(V_0)$ and $out(V_k) \subseteq out(V_0)$. $in(V_k)$ can now form the basis of a complete extension. The smallest of such (complete) extensions comes from what Caminada and Pigozzi called the *up-complete labelling* of λ_{V_k} :

Definition 2.19 ([7]). Let λ be an admissible labelling. The up-complete labelling of λ is a complete labelling λ' s.t. $in(\lambda') \supseteq in(\lambda)$ and $out(\lambda') \supseteq out(\lambda)$ and $in(\lambda')$ and $out(\lambda')$ are the smallest sets satisfying these conditions.

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

Definition 2.20. Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network, \mathbf{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 $V_e(X) = \lim_{i \rightarrow \infty} V_i(X)$.

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 $Att(X)$. 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. $\max_{Y \in Att(X)} \{V_k(Y)\} = 0$
2. $\max_{Y \in Att(X)} \{V_k(Y)\} = 1$

$$3. 0 < \max_{Y \in \text{Att}(X)} \{V_k(Y)\} < 1$$

If the value of a node Y at the stable point k is in $\{0, 1\}$, then Theorem 2.6 ensures that it will remain the same in the limit $\lim_{i \rightarrow \infty} V_i(Y)$. As it turns out, if $\max_{Y \in \text{Att}(X)} \{V_k(Y)\} = 0$, then $\lim_{i \rightarrow \infty} V_i(X) = 1$. And if $\max_{Y \in \text{Att}(X)} \{V_k(Y)\} = 1$, then $\lim_{i \rightarrow \infty} V_i(X) = 0$, as shown by the next theorem.

Theorem 2.21. *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation network and $V_0 : S \mapsto U$ assign initial values to the nodes in S . Let the sequence of value assignments V_0, V_1, V_2, \dots 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 $\max_{Y \in \text{Att}(X)} \{V_k(Y)\} = 0$, then $V_e(X) = 1$; and*
2. *If $\max_{Y \in \text{Att}(X)} \{V_k(Y)\} = 1$, then $V_e(X) = 0$.*
3. *If $V_k(X) \in \{0, 1\}$, then $V_e(X) = V_k(X)$;*

Proof. 1. If $\max_{Y \in \text{Att}(X)} V_k(Y) = 0$, and the sequence is stable at k , then by Corollary 2.8, $\max_{Y \in \text{Att}(X)} V_{k+j}(Y) = 0$, for all $j \geq 0$. We have that

$$\begin{aligned} V_{k+1}(X) &= (1 - V_k(X)) \cdot \min \left\{ \frac{1}{2}, 1 \right\} + V_k(X) \cdot \max \left\{ \frac{1}{2}, 1 \right\} \\ &= \frac{1}{2} - \frac{V_k(X)}{2} + V_k(X) = \frac{1}{2} + \frac{V_k(X)}{2} \\ V_{k+2}(X) &= \frac{1}{2} + \frac{1}{4} + \frac{V_k(X)}{4} \\ V_{k+j}(X) &= \sum_{k=1}^j \frac{1}{2^k} + \frac{V_k(X)}{2^j} \\ V_e(X) &= \lim_{j \rightarrow \infty} V_{k+j}(X) \\ &= \sum_{k=1}^{\infty} \frac{1}{2^k} + \lim_{j \rightarrow \infty} \frac{V_k(X)}{2^j} = 1 + 0 = 1 \end{aligned}$$

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 2.8, $\max_{Y \in \text{Att}(X)} V_{k+j}(Y) = 1$, for all $j \geq 0$. We have that

$$\begin{aligned} V_{k+1}(X) &= (1 - V_k(X)) \cdot \min \left\{ \frac{1}{2}, 0 \right\} + V_k(X) \cdot \max \left\{ \frac{1}{2}, 0 \right\} \\ &= \frac{V_k(X)}{2} \\ V_{k+2}(X) &= \frac{V_k(X)}{4} \quad \therefore V_{k+j}(X) = \frac{V_k(X)}{2^j} \\ V_e(X) &= \lim_{j \rightarrow \infty} V_{k+j}(X) = \lim_{j \rightarrow \infty} \frac{V_k(X)}{2^j} = 0 \end{aligned}$$

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

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

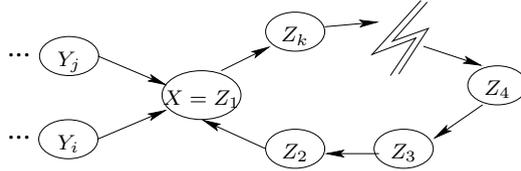


FIGURE 4. A network with a cycle with k nodes.

Theorem 2.22. *Let the sequence of values V_0, V_1, \dots , be stable at iteration k . Let X be a point such that $V_{k+i}(X), V_{k+i+1}(X), \dots \in (0, 1)$, for all $i \geq 0$. Our final aim is to show that $\lim_{i \rightarrow \infty} V_{k+i}(X) = \frac{1}{2}$. As a first step towards our goal, we show that any converging subsequence $V_{k+j}^c(X)$ converges to $\frac{1}{2}$ (by a subsequence $V_{k+j}^c(X)$ we mean some of the elements of the sequence $V_{k+i}(X)$, that is for every j there is an i_j such that $V_{k+j}^c(X) = V_{k+i_j}(X)$). From now on we talk about the subsequence $V_{k+j}^c(X)$, which we further assume that it converges to $V_e^c(X)$, for every such X , and we will show that $V_e^c(X) = \frac{1}{2}$, for every X .*

To be absolutely clear we assume for the time being that there is a sequence of values s_1, s_2, s_3, \dots , such that for every X , the sequence $V_{s_i}^c(X)$ converges to $V_e^c(X)$ and we show that under these conditions $V_e^c(X) = \frac{1}{2}$.

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

Proof. Since the Gabbay-Rodrigues Iteration Schema uses continuous functions, if the schema holds for the elements of the sequence $V_{k+j}^c(X)$, for every $X \in S$, it also holds for the limit $V_e^c(X)$.

We get the following systems of equations

1. For the cycle $X = Z_1 \leftarrow Z_2 \leftarrow \dots \leftarrow Z_{2n} = X$:

$V_e^c(X) = (1 - V_e^c(X)) \cdot \min\{\frac{1}{2}, 1 - V_e^c(Y)\} + V_e^c(X) \cdot \max\{\frac{1}{2}, 1 - V_e^c(Y)\}$,
where Y is the node in $Att(X)$ with maximum value. We have two cases to consider.

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

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

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

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

it is easy to see from the equations that if $V_e^c(Y) \geq \frac{1}{2}$, then $V_e^c(X) \leq \frac{1}{2}$ and if $V_e^c(Y) \leq \frac{1}{2}$, then $V_e^c(X) \geq \frac{1}{2}$. Therefore, if we have the cycle $X = Z_1 \leftarrow Z_2 \leftarrow \dots \leftarrow Z_{2n} = X$, 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 $X = Z_1 \leftarrow Z_2 \leftarrow \dots \leftarrow Z_{2n+1} = X$, we have that

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

$$\begin{aligned} V_e^c(X) &= \frac{1 - V_e^c(Y)}{1.5 - V_e^c(Y)} \\ &= \frac{1 - \frac{1}{2} - \epsilon(Y)}{1.5 - (\frac{1}{2} + \epsilon(Y))} \\ &= \frac{\frac{1}{2} - \epsilon(Y)}{1 - \epsilon(Y)} \end{aligned}$$

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

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

$$\begin{aligned} \eta &= \frac{1}{2} - \frac{\frac{1}{2} - \epsilon(Y)}{1 - \epsilon(Y)} \\ &= \frac{(1\epsilon(Y)) - 2(\frac{1}{2} - \epsilon(Y))}{2(1 - \epsilon(Y))} \\ &= \frac{1 - \epsilon(Y) - 1 + 2\epsilon(Y)}{2(1 - \epsilon(Y))} \\ &= \frac{\epsilon(Y)}{2(1 - \epsilon(Y))} \end{aligned}$$

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

$$\begin{aligned}
 V_e^c(X) &= \frac{1}{1 + 2(\frac{1}{2} - \epsilon(Y))} \\
 &= \frac{1}{1 + 1 - 2\epsilon(Y)} \\
 &= \frac{1}{2(1 - \epsilon(Y))} \\
 &= \frac{1}{2} + \eta
 \end{aligned}$$

$$\begin{aligned}
 \eta &= \frac{1}{2(1 - \epsilon(Y))} - \frac{1}{2} \\
 &= \frac{1 - 1 + \epsilon(Y)}{2(1 - \epsilon(Y))} \\
 &= \frac{\epsilon(Y)}{2(1 - \epsilon(Y))}
 \end{aligned}$$

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

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

Let us now assume a cycle

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

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

$$\begin{aligned}
\eta_{k+1} &= \frac{\eta_k}{2(1 - \eta_k)} \\
&= \frac{\frac{\epsilon}{2(2^k - (2^k - 1)\epsilon)}}{2\left(1 - \frac{\epsilon}{2(2^k - (2^k - 1)\epsilon)}\right)} \\
&= \frac{\frac{\epsilon}{2(2^k - (2^k - 1)\epsilon)}}{2\left(\frac{2(2^k - (2^k - 1)\epsilon - \epsilon)}{2(2^k - (2^k + 1)\epsilon)}\right)} \\
&= \frac{\epsilon}{2(2^{k+1} - 2^{k+1}\epsilon + 2\epsilon - \epsilon)} \\
&= \frac{\epsilon}{2(2^{k+1} - (2^{k+1} - 1)\epsilon)}
\end{aligned}$$

So the recursion works. Now if we have a loop, we get

$$Z_n = Z_1$$

So $\eta_n = \eta_1$ and thus

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

If we divide by η ($\neq 0$), we get

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

It is easy to see that only $\epsilon = \frac{1}{2}$ solves the equation. This means that $V_e^c(Z_i) = \frac{1}{2}$, for all Z_i .

□

Remark 2.23. 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 $V_{k'}^c$ 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 $Z_2^{\alpha_2}$ be one arbitrary such choice. Repeating this consideration now for $Z_2^{\alpha_2}$ and for the subsequence thus obtained, we get a $Z_3^{\alpha_3}$ and a further subsequence of the subsequence and so on. Eventually, we get a final subsequence (which depends on the choices of $Z_i^{\alpha_i}$) $V_{k+r_1}^c, V_{k+r_1+r_2}^c, \dots$, such that $Z_{i+1}^{\alpha_{i+1}}$ is the node in $\text{Att}(Z_i^{\alpha_i})$ with maximum value and $0 < V_{k+r_1+r_2+\dots+r_m}^c(Z_{i+1}^{\alpha_{i+1}}) < 1$, for each m .

Remark 2.24. We use a similar argument to the one in Remark 2.23 to show that if a subsequence $V_{k+j}^c(X)$ converges to $V_e^c(X)$, then it can be further refined to a subsequence $V_{s_i}^c$ such that $V_{s_i}^c(Y)$ converges for all Y . The reason is that the number of such Y is finite (since S is finite). We can then successively refine the sequence $V_{k+j}^c(X)$ into subsequences for which $V_{k+j}^c(Y)$ also converges. Therefore, Theorem 2.22, can be used to show that the convergent

sequence V_{k+j}^c converges to $\frac{1}{2}$. We can therefore further conclude that every convergent subsequence of $V_{k+m}(X)$ converges to $\frac{1}{2}$. The next lemma shows that the sequence $V_{k+m}(X)$ itself converges to $\frac{1}{2}$.

Lemma 2.25. *Let $\alpha = \alpha_1, \alpha_2, \alpha_3, \dots$, be an infinite sequence of values in $[0, 1]$. If every convergent subsequence of α converges to $\frac{1}{2}$, then $\lim_{i \rightarrow \infty} \alpha_i = \frac{1}{2}$.*

Proof. For every $0 < \varepsilon < \frac{1}{2}$, $[\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon]$ only a finite number of α_i 's are in $[0, \frac{1}{2} - \varepsilon] \cup [\frac{1}{2} + \varepsilon, 1]$. Otherwise, say $[0, \frac{1}{2} - \varepsilon]$ has an infinite number of α_i 's. Then since $[0, \frac{1}{2} - \varepsilon]$ is a closed interval with an infinite number of values in it, there would exist an infinite convergent subsequence of α in it that does not converge to $\frac{1}{2}$.

Therefore, we have shown that for every $0 < \varepsilon < \frac{1}{2}$, ε small, there exists a number m such that for every $n > m$, $(\frac{1}{2} - \alpha_n) \in [\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon]$, that is $\lim_{i \rightarrow \infty} \alpha_i = \frac{1}{2}$. \square

Theorem 2.21 asserts what the limit values of the nodes whose values of the attackers are known at the stable iteration k . Theorem 2.26 asserts the same in terms of the *limit* values of the attackers.

Theorem 2.26.

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 $\lim_{j \rightarrow \infty} \{V_{j+1}(X)\} = \lim_{j \rightarrow \infty} \{V_j(X)\}$.

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

$$\begin{aligned} V_e(X) &= (1 - V_e(X)) \cdot \min \left\{ \frac{1}{2}, 1 \right\} + V_e(X) \cdot \max \left\{ \frac{1}{2}, 1 \right\} \\ V_e(X) &= (1 - V_e(X)) \cdot \frac{1}{2} + V_e(X) \\ 2 \cdot V_e(X) &= 1 - V_e(X) + 2 \cdot V_e(X) \\ V_e(X) &= 1 \end{aligned}$$

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

$$\begin{aligned} V_e(X) &= (1 - V_e(X)) \cdot \min \left\{ \frac{1}{2}, 0 \right\} + V_e(X) \cdot \max \left\{ \frac{1}{2}, 0 \right\} \\ V_e(X) &= \frac{V_e(X)}{2} \\ V_e(X) &= 0 \end{aligned}$$

So if the equilibrium value of any of the attackers of X is 1, then the equilibrium value of X is 0. \square

Theorem 2.27. *Let $\langle S, R \rangle$ be an argumentation network and \mathbf{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, \dots , where each $V_i, i > 0$, is generated by \mathbf{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 $V_1(X) = \min\{\frac{1}{2}, 0\} = 0$. 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, $V_1(X) = \max\{\frac{1}{2}, 1\} = 1$.

The stability of the crisp values then follows from Theorem 2.6 and since $0 < V_0(X) < 1$, then by Theorem 2.4 (case 3), so does the stability of the remaining non-crisp values. \square

Proposition 2.28. *Let $\langle S, R \rangle$ be an argumentation network; \mathbf{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 λ be a legal labelling function.*

Take any $X \in S$. If λ and V_e agree on the values of all nodes in $\text{Att}(X)$, then λ 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 2.21.

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 $V_e(X) = \sum_{k=1}^{\infty} \frac{1}{2^k} + \lim_{j \rightarrow \infty} \frac{V_k(X)}{2^j} = 1 + 0 = 1$. Since V_e and λ agree with each other on the values of all nodes in $\text{Att}(X)$, we have that for all $Y \in \text{Att}(X)$, $\lambda(Y) = \mathbf{out}$ and since λ is legal, $\lambda(X) = \mathbf{in}$, and hence λ 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 $V_e(X) = \lim_{j \rightarrow \infty} \frac{V_e(X)}{2^j} = 0$. Since V_e and λ agree with each other on the values of all nodes in $\text{Att}(X)$, we have that $\lambda(Y) = \mathbf{in}$ and since λ is legal, $\lambda(X) = \mathbf{out}$. Hence λ 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 $\lambda(Y) = \mathbf{und}$) and for no $Y \in \text{Att}(X)$, $V_e(Y) = 1$ (and hence for no $Y \in \text{Att}(X)$, $\lambda(Y) = \mathbf{in}$). 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 λ is legal, $\lambda(X) = \mathbf{und}$, and hence λ and V_e agree with each other with respect to the value of X . \square

And now to the main theorem of this section, which explains the equilibrium values of all nodes and shows their relationship to Caminada and

Pigozzi's down-admissible/up-complete constructions. A down-admissible labelling is obtained after a series of *contraction operations* as defined below.

Definition 2.29 ([7]). *Let λ be a labelling of an argumentation network $\langle S, R \rangle$. A contraction sequence from λ is a sequence of labellings $[\lambda_1 = \lambda, \dots, \lambda_k]$ such that*

1. *For each $i \in \{1, \dots, k-1\}$, $\lambda_{i+1} = \lambda_i - \{(X, \mathbf{in}), (X, \mathbf{out})\} \cup \{(X, \mathbf{und})\}$, where X is an argument that is illegally labelled **in**, or illegally labelled **out** in λ_i ; and*
2. *λ_k is a labelling without any arguments illegally labelled **in** or illegally labelled **out**.*

Theorem 6 of [7] shows us that if we successively contract an initial labelling λ , then at the end of the contraction sequence $[\lambda_1 = \lambda, \lambda_2, \dots, \lambda_k]$, λ_k corresponds to the down-admissible labelling of λ , which is the largest admissible labelling that is smaller or equal to λ .

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

Definition 2.30 ([7]). *Let λ be an admissible labelling of the argumentation network $\langle S, R \rangle$. An expansion sequence from λ is a sequence of labellings $[\lambda_1 = \lambda, \dots, \lambda_k]$ such that*

1. *For each $i \in \{1, \dots, k-1\}$,*

$$\lambda_{i+1} = \begin{cases} \lambda_i - \{(X, \mathbf{und})\} \cup \{(X, \mathbf{in})\}, & \text{if } X \text{ is an argument that is} \\ & \text{illegally labelled } \mathbf{und} \text{ in } \lambda_i \text{ and all its attackers are labelled } \mathbf{out} \\ \lambda_i - \{(X, \mathbf{und})\} \cup \{(X, \mathbf{out})\}, & \text{if } X \text{ is an argument that is} \\ & \text{illegally labelled } \mathbf{und} \text{ in } \lambda_i \text{ and it has an attacker labelled } \mathbf{in} \end{cases}$$
2. *λ_k is a labelling without any arguments illegally labelled **und**.*

Caminada and Pigozzi have shown us that if $[\lambda_1 = \lambda, \dots, \lambda_k]$ is an expansion sequence,¹⁵ then λ_k is a complete labelling and it is the smallest such labelling containing λ . We now introduce a few concepts to help us in the proof of our main theorem.

Definition 2.31. *Let $\langle S, R \rangle$ be an argumentation network; V be an assignment of values to the nodes in S ; and λ a labelling of these nodes. We say that V and λ 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 $\lambda(X) = \mathbf{in}$
2. $V(X) = 0$ if and only if $\lambda(X) = \mathbf{out}$
3. $V(X) = \frac{1}{2}$ if and only if $\lambda(X) = \mathbf{und}$

¹⁵Note λ_1 must be admissible.

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

Definition 2.32 (Attack tree of a node). Let $\langle S, R \rangle$ be a network. The attack tree $\text{tree}(X)$ 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 $\text{Att}(N)$.

Definition 2.33 (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 $X = Z_0, Z_1, Z_2, \dots$ 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 $\Pi(X)$. We allow for a single node to be a path.

Using paths, we can define a *strongly connected component* (SCC) to be a maximal subset $C \subseteq S$, such that for every $X, Y \in C$, there exists a path from X containing Y .

Note that in a SCC C for every path $\pi = Z_0, Z_i, \dots$ 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 π that is involved in a loop, or you can think of it as the minimum distance from the starting node of the path π 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 $\pi = Z_0, Z_1, \dots$, the node $Z_{i(\pi)}$.

Definition 2.34 (V_{\max} -paths). Let Z be a node in a SCC C and let the sequence of values V_0, V_1, \dots be stable at iteration k . The set of V_{\max} -paths of Z is defined as $V_{\max}\text{-paths}(Z) = \{\pi = [Z = Z_0, Z_1, \dots] \in \Pi(Z) \mid \text{for each } Z_i, V_{k+r}(Z_{i+1}) = \max_{Z'_{i+1}} \{V_{k+r}(Z'_{i+1})\} \text{ for an infinite number of } r\}$'s.

For every $Z \in C$, the set of V_{\max} -paths from Z is non-empty (see Remark 2.23).

Definition 2.35 (Bar of a node). Let C be a SCC and take $X \in C$. The bar of X is the set

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

Definition 2.36. Let $\Gamma(X)$ be the set of V_{\max} -paths of X and take $U \subseteq C$ a set of points. The bar of X modified by U is defined as

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

Theorem 2.37. Let $\langle S, R \rangle$ be an argumentation network; V_0 be an initial assignment of values to the nodes in S ; λ_0 an initial labelling of these nodes; and V_0 and λ_0 faithful to each other according to Definition 2.16. Let λ_{da} be the labelling at the end of a contraction sequence from λ_0 and λ_{CP} the labelling at the end of an expansion sequence after λ_{da} . Let k be the point at which the sequence V_0, V_1, \dots becomes stable and $V_e(X)$ the equilibrium value of a node calculated through the Gabbay-Rodrigues Iteration Schema. Then λ_{CP} and V_e agree with each other according to Definition 2.31.

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 $\max_{Y \in Att(X)} V_0(Y) = \max_{Y \in Att(X)} V_k(Y) = 0$ and then by Theorem 2.21, $V_e(X) = 1$.

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

Case 2: X is part of a source SCC C and both $V_0 \upharpoonright C$ and $\lambda_0 \upharpoonright C$ 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 $\lambda_0 \upharpoonright C$ is a legal assignment, and the nodes in C^c only have values in $\{\mathbf{in}, \mathbf{out}\}$, 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 $\lambda_{CP} \upharpoonright C^c = \lambda_0 \upharpoonright C^c$. By construction, the values of all nodes in C^u are \mathbf{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 $\lambda_{CP} \upharpoonright C = \lambda_0 \upharpoonright C$.

Since $V_0 \upharpoonright C$ is a legal assignment, then by Theorem 2.27, it is stable at iteration 0. As a result, for all nodes $X \in C^c$, $V_1(X) = V_0(X)$. Hence by Theorem 2.6, $V_e(X) = V_0(X)$ for all nodes $X \in C^c$, and then since λ_0 and V_0 are faithful to each other (Definition 2.16), conditions 1. and 2. of Definition 2.31 are satisfied. We now show that condition 3. also follows. For all nodes $X \in C^u$, we have that $0 < V_0(X) < 1$. Since $V_0 \upharpoonright C$ is a legal assignment, then for every $X \in C^u$, $0 < \max_{Y \in Att(X)} \{V_0(Y)\} < 1$.¹⁶ Notice that by construction $C^u = C \setminus C^c$. Stage two of case 3 below shows that for all nodes $X \in C^u$, $V_e(X) = \frac{1}{2}$. Therefore, condition 3. of Definition 2.31 is also satisfied and as a result, λ_{CP} and V_e agree with each other with respect to all nodes in C .

case 3: X is part of a source SCC C and $\lambda_0 \upharpoonright C$ and $V_0 \upharpoonright C$ are not legal assignments.

Stage one:

We know that the sequence of assignments V_0, V_1, \dots , eventually becomes stable at some iteration k and by Theorem 2.14, $in(V_k) \subseteq$

¹⁶This effectively means that the only possible incoming attacks from C^c are from nodes labelled \mathbf{out} . Otherwise, the attacked nodes in C^u should have been labelled \mathbf{out} and hence would have been illegally labelled \mathbf{und} .

$in(V_0)$, $out(V_k) \subseteq out(V_0)$ and $in(V_k)$ is the largest admissible subset of $in(V_0)$. By Theorem 6 of [7], $in(\lambda_{CP})$ is the largest (and unique) admissible subset of $in(\lambda_0)$ and since λ_0 and V_0 are faithful to each other, we can conclude that $in(V_k) = in(\lambda_{da})$ and $out(V_k) = out(\lambda_{da})$.

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

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

Suppose that e_1 turns the node X illegally labelled \mathbf{und} by λ_{da} into \mathbf{in} . Then $out(\lambda_1) = out(\lambda_{da})$ and $in(\lambda_1) = in(\lambda_{da}) \cup \{X\}$. Then for all $Y \in Att(X)$, $\lambda_{da}(Y) = \mathbf{out}$. Therefore, $V_k(Y) = 0$ for all $Y \in Att(X)$, and hence $\max_{Y \in Att(X)} \{V_k(Y)\} = 0$. By Theorem 2.21, $V_e(X) = 1$ and therefore $X \in in(V_e)$. We set $V_k^{1,out} = out(V_k)$ and $V_k^{1,in} = in(V_k) \cup \{X\}$.

Suppose that e_1 turns the node X illegally labelled \mathbf{und} by λ_{da} into \mathbf{out} . Then $in(\lambda_1) = in(\lambda_{da})$ and $out(\lambda_1) = out(\lambda_{da}) \cup \{X\}$. Then there exists $Y \in Att(X)$ such that $\lambda_{da}(Y) = \mathbf{in}$. Therefore, $V_k(Y) = 1$ for some $Y \in Att(X)$, and hence $\max_{Y \in Att(X)} \{V_k(Y)\} = 1$. By Theorem 2.21, $V_e(X) = 0$ and therefore $X \in out(V_e(X))$. We set $V_k^{1,out} = out(V_k) \cup \{X\}$ and $V_k^{1,in} = in(V_k)$.

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

Suppose that e_{i+1} turns the node X illegally labelled \mathbf{und} in λ_i into \mathbf{in} . Then for all $Y \in Att(X)$, $\lambda_i(Y) = \mathbf{out}$. Therefore, $V_e(Y) = 0$ for all $Y \in Att(X)$, and hence $\max_{Y \in Att(X)} \{V_e(Y)\} = 0$. By Theorem 2.26, $V_e(X) = 1$ and therefore $X \in in(V_e)$. As before, we set $V_k^{i+1,out} = V_k^{i,out}$ and $V_k^{i+1,in} = in(V_k) \cup \{X\}$.

Suppose that e_{i+1} turns the node X illegally labelled \mathbf{und} by λ_i into \mathbf{out} . Then there exists $Y \in Att(X)$ such that $\lambda_i(Y) = \mathbf{in}$. Therefore, $V_e(Y) = 1$ for some $Y \in Att(X)$, and hence $\max_{Y \in Att(X)} \{V_e(Y)\} = 1$. By Theorem 2.26, $V_e(X) = 0$ and therefore $X \in out(V_e(X))$. Again, we set $V_k^{i+1,out} = V_k^i \cup \{X\}$ and $V_k^{i+1,in} = V_k^{i,in}$.

By now we know that if $X \in V_k^{m,in}$, then $V_e(X) = 1$ and $\lambda_{CP}(X) = \mathbf{in}$ and that $X \in V_k^{m,out}$, then $V_e(X) = 0$ and $\lambda_{CP}(X) = \mathbf{out}$. We ask if there is some $Z \notin V_k^{m,in}$ such that $V_e(Z) = 1$ or $Z \notin V_k^{m,out}$ 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 $(V_k^{m,in} \cup V_k^{m,out})$ and C^u to denote $C \setminus C^c$. Suppose $X \in C^u$.

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

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

So we know that for all $X \in C^u$, $\lambda_{CP}(X) = \mathbf{und}$ 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 $X = Z_0, Z_1, Z_2, \dots, Z_k = X$, where for every $i > 0$, $(Z_{i+1}, Z_i) \in R$. Some of the Z_i are in $V_k^{m,out}$, but none can be in $V_k^{m,in}$, for that would make Z_{i-1} **out**.

The proof is done by induction on the maximum distance from a node X in C^u to a loop $Z_1, Z_2, \dots, Z_k = Z_1$, 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 $\pi(X) = (Z_0 = X), Z_1, \dots$. Now define the distance of X , $\dim X$, as the maximum index i such that for each path $\pi(X)$, $Z_i \in bar(Z, V_k^{m,out})$. 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 $V_k^{m,out}$.

If $\dim X = 0$, then X must be attacked by a cycle involving only X (otherwise $X \in V_k^{m,out}$, 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 2.22, 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

$$\begin{aligned}
 V_e(X) &= (1 - V_e(X)) \cdot \min \left\{ \frac{1}{2}, \frac{1}{2} \right\} + V_e(X) \cdot \max \left\{ \frac{1}{2}, \frac{1}{2} \right\} \\
 &= \frac{1 - V_e(X)}{2} + \frac{V_e(X)}{2} \\
 &= \frac{1 - V_e(X) + V_e(X)}{2} \\
 &= \frac{1}{2}
 \end{aligned}$$

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 $Y \in Att(X)$, we have that $\dim Y \leq k$. Therefore, either $Y \in V_k^{m,out}$ in which case

$V_e(Y) = 0$, or by the inductive hypothesis $V_e(Y) = \frac{1}{2}$.¹⁷ Therefore we have that $\max Y \in \text{Att}(X)\{V_e(Y)\} = \frac{1}{2}$ and as before

$$\begin{aligned} V_e(X) &= (1 - V_e(X)) \cdot \min \left\{ \frac{1}{2}, \frac{1}{2} \right\} + V_e(X) \cdot \max \left\{ \frac{1}{2}, \frac{1}{2} \right\} \\ &= \frac{1}{2} \end{aligned}$$

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

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

Define $\text{Known}_{k+1}^0 = \{X \in S \mid \text{depth}(X) \leq k\}$ and $\text{Known}_{k+1}^{m+1} = \{X \in S \mid \text{depth}(X) = k + 1 \text{ and for all } Y \in \text{Att}(X), Y \in \text{Known}_{k+1}^m\}$.

We show that for all $i \geq 0$, we have that $\lambda_{CP}(X) = V_e(X)$, for all $X \in \text{Known}_{k+1}^i$. First notice that by induction hypothesis, $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_{k+1}^0$. Now suppose that $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_{k+1}^i$, then by Proposition 2.28, $\lambda_{CP}(X) = V_e(X)$ for all $X \in \text{Known}_{k+1}^{i+1}$. Since the network is finite, $\text{Known}_{k+1}^e = \text{Known}_{k+1}^{e+1}$, for some $e \geq 0$. Define $C_{k+1}^u = \{X \in S \mid \text{depth}(X) = k + 1\} \setminus \text{Known}_{k+1}^e$.

By definition, if there exists $X \in C_{k+1}^u$ and $Y \in \text{Att}(X)$ such that $Y \in \text{Known}_{k+1}^e$, then $\lambda_{CP}(Y) = \mathbf{out}$ and $V_e(Y) = 0$ (otherwise the value of X would be known). Therefore, we can exclude the nodes in Known_{k+1}^e 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 $X \in C_{k+1}^u$, $V_e(X) = \lambda_{CP}(X)$. \square

Corollary 2.38. *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 $X \in S$, $V_e(X) \in \{0, \frac{1}{2}, 1\}$.*

Proof. Follows from the possible equilibrium values of all nodes in Theorem 2.37. \square

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

¹⁷Note that $\text{Att}(X) \not\subseteq V_k^{m,out}$, otherwise X would be illegally labelled **und**.

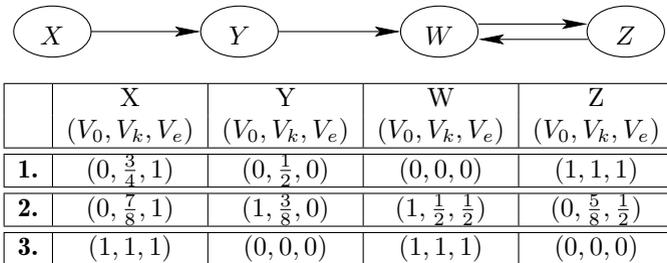


FIGURE 5. Network used in Section 3.

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 their resulting up-completion according to Caminada-Pigozzi's procedure can be obtained simply by replacing 0 with **out**, 1 with **in** and values in $(0, 1)$ with **und**.

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 $E_1 = \{X, Z\}$. Contrast this with case **2.**, in which the initial values of W and Z are 1 and 0, resp. The extension $E = \{X, W\}$ is also complete but is obtained neither by our procedure nor by Caminada-Pigozzi's down-admissible/up-complete construction. This can be explained as follows. The initial illegal value of Y invalidates the initial acceptance of W , turning it into undecided in the calculation of the down-admissible subset. From that point on, the original legal assignments for W and Z can no longer be restored and they both end up as undecided. As a result, we obtain the complete (but not preferred) extension $E_2 = \{X\}$. This interference does not happen in case **1.**, because there the interference of the undecided value of Y over W is dominated by Z 's 1 value that keeps W 's 0 value in check (because of the behaviour of max). As a result, both W 's and Z 's initial values are retained.

If however we start with a preferred extension, which is also complete by definition, we get as a result unchanged initial values (cf. Theorem 2.37). Caminada-Pigozzi also give the same result because the down-admissible labelling of a labelling yielding a preferred extension is the labelling itself and since that labelling is also complete, then the up-completion does not change anything (case **3.** in the table of Figure 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 (*Crisp*) and a set of initial values to the nodes.

1. Calculate the equilibrium values for all nodes using the iteration schema.
2. If $\{X \in S \mid V_e(X) \in \{0, 1\}\} \subseteq \text{Crisp}$, stop. The extension is defined in the set $\{X \mid V_e(X) = 1\}$. Otherwise, set $\text{Crisp} = \text{Crisp} \cup \{X \in S \mid V_e(X) \in \{0, 1\}\}$ 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 *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 $\frac{1}{2}$. The procedure also *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 $V_e(X) = 1$, $V_e(Y) = 0$, $V_e(W) = V_e(Z) = \frac{1}{2}$. Hence, $Crisp = \{X, Y\}$. 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 $Crisp = \{X, Y, W, Z\}$. 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.

Obviously, the procedure can also be applied using Caminada-Pigozzi's construction instead of the Gabbay-Rodrigues Iteration Schema of step 1. above.

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 row of the table indicates the iteration in which the stable values were reached and the equilibrium values approximated (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.¹⁸ 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 $\frac{1}{2}$ 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 interference of the undecidedness of B over X is dominated by the Y 's value of 1 and the initial values of both X and Y persist. However, if X starts with 1 and Y with 0, the undecidedness of B will then "contaminate" the X - Y loop. It will force X

¹⁸Effectively this means that if the maximum variation in node values between two successive iterations is smaller than 10^{-19} , 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.

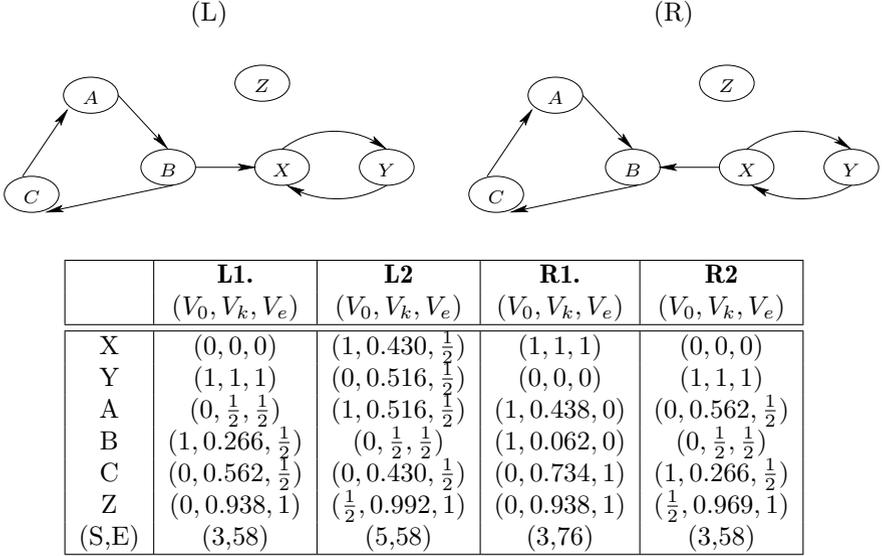


FIGURE 6. Equilibrium and stable values of nodes involved in cycles.

to become undecided, which in turn makes Y also become undecided. As a result, all of the values will converge to $\frac{1}{2}$ apart from Z 's, which as we said is independent and will converge 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 $\frac{1}{2}$ independently of their initial values (R2).

4. 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 were in the context of the equational approach to an argumentation network when we are given some initial values. Our problem was to find a solution to the system of equations that was “close” to these initial values.

Two important concepts which are directly related to the work presented in this paper were proposed in [7], which addressed the problem of finding an extension of an argumentation network given an initial labelling

of its arguments. Their procedure works in two steps. Firstly, they calculate the downward-admissible labelling of the original labelling, which essentially consists of an admissible labelling whose crisp part is maximally included in the original labelling. This is done by a procedure which at each step, turns an illegally labelled argument from **in** or **out** into **und** until no illegal crisp values remain. They called this step a *contraction sequence* and it is similar to what our schema does to the sequence of value assignments until it becomes stable, except that at each iteration our schema may contract more than one node simultaneously, whereas theirs contracts only one node per iteration. More importantly, their procedure is *non-deterministic*: it selects an illegally labelled node for contraction, but this requires searching for such nodes. Hence there is an implicit cost involved in it. Even though the search can be optimised, it renders the overall cost of the procedure in terms of steps higher than ours, which is truly bounded by $|S|$. Now, given an admissible labelling, a complete extension is constructed by turning nodes that are illegally labelled **und** into **in** or **out** as appropriate. They call this step an *expansion* and its counterpart in our procedure is the calculation of the limit values of the sequence. Obviously, in a computer program, we can only approximate these limit values. In our implementation, we stop the iterations when we can no longer guarantee the accuracy of the calculations without introducing rounding errors due to the limitations of the processor. This happens in linear time too (see Figure 6). In practice, the limit values can be guessed much earlier as the iteration values can be seen to be converging towards one of the three values 0, $\frac{1}{2}$ and 1.

We stress that neither are we limited to the discreet values **out**, **in** and **und**, nor to the Eq_{\max} equation used in the iteration schema and this allows the application of the schema in the calculation of extensions given different semantics (see Section 5).

One can take a different approach to the one 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, $\langle S, R \rangle$ is an acyclic argumentation network and $f : S \mapsto U$ is a function assigning initial values to the nodes in S .

Definition 4.1. Consider the sequence $\alpha_0(X), \alpha_1(X), \dots, \alpha_i(X), \dots$, where

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

and let

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

Definition 4.2. The attack depth of a node X of an acyclic argumentation network, in symbols $a\text{-depth}(X)$, is defined recursively as

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

The function $a\text{-depth}$ is well-defined, because there are no cycles in $\langle S, R \rangle$.

Definition 4.3. Given initial values for the nodes of an acyclic network, the function $\beta : S \mapsto U$ provides a means of calculating fixed-point values for all nodes as follows.

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

Theorem 4.4. $\alpha(X) = \beta(X)$ 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

$$\begin{aligned}\alpha_0(X) &= f(X) \\ \alpha_1(X) &= \frac{1}{2} \alpha_0(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y) \right\} \\ &= \frac{1}{2} f(X) + \frac{1}{2} f(X) \\ &= f(X) \\ \alpha_2(X) &= \frac{1}{2} \alpha_1(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_1(Y) \right\} \\ &= \frac{1}{2} f(X) + \frac{1}{2} f(X) = f(X) \\ \alpha(X) &= \lim_{i \rightarrow \infty} \left\{ \frac{1}{2} \alpha_i + \frac{1}{2} f(X) \right\} \\ \alpha(X) &= f(X) = \beta(X)\end{aligned}$$

(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 $f(Y) = \alpha_0(Y) = \alpha_1(Y) = \alpha_i(Y) = \dots = \alpha(Y) = \beta(Y)$.

$$\begin{aligned}
\alpha_0(X) &= f(X) \\
\alpha_1(X) &= \frac{1}{2}\alpha_0(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y) \right\} \\
&= \frac{1}{2}f(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
\alpha_2(X) &= \frac{1}{2} \left(\frac{1}{2}f(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \right) + \\
&\quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
&= \frac{1}{2^2}f(X) + \frac{1}{2^2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} + \\
&\quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
\alpha_i(X) &= \frac{1}{2^i}f(X) + \sum_{j=1}^i \frac{1}{2^j} \cdot \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
&= \frac{1}{2^i}f(X) + \left(1 - \frac{1}{2^i} \right) \cdot \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
\alpha(X) &= \lim_{i \rightarrow \infty} \alpha_i(X) \\
&= \lim_{i \rightarrow \infty} \frac{1}{2^i}f(X) + \left(1 - \frac{1}{2^i} \right) \cdot \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
&= \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \beta(Y) \right\} \\
&= \beta(X)
\end{aligned}$$

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

$$\begin{aligned}
\alpha_0(X) &= f(X) \\
\alpha_1(X) &= \frac{1}{2}\alpha_0(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y) \right\} \\
&= \frac{1}{2}f(X) + \\
&\quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y) \right\} \\
\alpha_2(X) &= \frac{1}{2} \left(\frac{1}{2}f(X) + \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in \text{Att}(X)} \alpha_0(Y) \right\} \right) +
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_1(Y) \right\} \\
&= \frac{1}{2^2} f(X) + \frac{1}{2^2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_0(Y) \right\} + \\
& \quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_1(Y) \right\} \\
\alpha_3(X) &= \frac{1}{2} \left(\frac{1}{2^2} f(X) + \frac{1}{2^2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_0(Y) \right\} + \right. \\
& \quad \left. \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_1(Y) \right\} \right) + \\
& \quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_2(Y) \right\} \\
&= \frac{1}{2^3} f(X) + \frac{1}{2^3} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_0(Y) \right\} + \\
& \quad \frac{1}{2^2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_1(Y) \right\} + \\
& \quad \frac{1}{2} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_2(Y) \right\} \\
\alpha_i(X) &= \frac{1}{2^i} f(X) + \frac{1}{2^{i-0}} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_0(Y) \right\} + \\
& \quad \frac{1}{2^{i-1}} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_1(Y) \right\} + \dots + \\
& \quad \frac{1}{2^1} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_{i-1}(Y) \right\} \\
\alpha_{i+1}(X) &= \frac{1}{2^i} f(X) + \sum_{i=1}^i \frac{1}{2^i} \cdot \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_i(Y) \right\} \\
&= \frac{1}{2^i} f(X) + \left(1 - \frac{1}{2^i} \right) \cdot \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_i(Y) \right\} \\
\alpha(X) &= \lim_{i \rightarrow \infty} \frac{1}{2^i} f(X) + \left(1 - \frac{1}{2^i} \right) \cdot \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_i(Y) \right\} \\
&= \lim_{i \rightarrow \infty} \left(1 - \frac{1}{2^i} \right) \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_i(Y) \right\} \\
&= \lim_{i \rightarrow \infty} \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha_i(Y) \right\} \\
&= \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \lim_{i \rightarrow \infty} \alpha_i(Y) \right\} \\
\alpha(X) &= \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \alpha(Y) \right\}
\end{aligned}$$

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

$$\alpha(X) = \min \left\{ f(X), 1 - \max_{Y \in Att(X)} \beta(Y) \right\} = \beta(X)$$

□

The theorem above 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 4.3.

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).

5. 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 Prakken's work [20]; or they can be obtained as the result of the merging of several networks, as proposed in [17, 16].

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:

$$V_{i+1}(X) = (1 - V_i(X)) \cdot \min \left\{ \frac{1}{2}, g(\mathcal{N}(X)) \right\} + V_i(X) \cdot \max \left\{ \frac{1}{2}, g(\mathcal{N}(X)) \right\}$$

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).¹⁹ Other operations can be used for argumentation systems, whose relationship with the schema is being further investigated. One such operation is *product*, which unlike min combines the strength of the attacks on a node. Another interesting possibility is to use the schema for *abstract dialectical frameworks* (ADFs) [3]. ADFs require the specification of a possibly unique type of equation for each node. Consider the ADF with nodes a, b, c and d with $R = \{(a, b), (b, c), (c, c)\}$. The ADF equations are: $C_a = \top$, $C_b = a$, $C_c = c \wedge b$ and $C_d = \neg d$. The complete models for this ADF are $m_1 = (t, t, u, u)$, $m_2 = (t, t, t, u)$ and $m_3 = (t, t, f, u)$. The Gabbay-Rodrigues schema converges to m_1 given initial values $(1, 1, \frac{1}{2}, \frac{1}{2})$; 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 [7], in time t 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. Obviously, the values corresponding to a legitimate extension are all legal. If they are given as input, the sequence will immediately stabilise. 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 can thus be preserved and hence we can end up with a larger complete extension than the one obtained through a single run. This extension is as compatible as possible with the initial values.

Acknowledgements

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

¹⁹Note that $1 - \max_{Y \in Att(X)} \{V(Y)\} = \min_{Y \in Att(X)} \{1 - V(Y)\}$.

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. Eğılmez, 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. Gabbay and O. Rodrigues. Probabilistic argumentation. An equational approach. To appear.
- [13] D. M. Gabbay. Introducing equational semantics for argumentation networks. DOI: 10.1007/978-3-642-22152-1_2, 2011.
- [14] D. M. Gabbay. Equational approach to argumentation networks. *Argument and Computation*, 3:87–142, 2012. DOI: 10.1080/19462166.2012.704398.
- [15] 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.
- [16] D. M. Gabbay and O. Rodrigues. A equational approach to the merging of argumentation networks. *Journal of Logic and Computation*, 2012.
- [17] 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.

- [18] M. P. Hassell. *The Dynamics of Arthropod Predator-Prey Systems*. Princeton University Press, 1978.
- [19] J. Leite and J. Martins. Social abstract argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011. To appear.
- [20] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
- [21] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, September 2003.

Appendix 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 an example from biology. This is a model by M. P. Hassell [18] of the dynamics of a system with two parasitoids (\mathbf{P} and \mathbf{Q}) and one host (\mathbf{N}). The interactions in the ecology are depicted in Figure 7. The equations modelling the dynamics are the following (see [1, p. 295]).*

$$\begin{aligned} N_{t+1} &= \lambda N_t f_1(\mathbf{P}_t) f_2(\mathbf{Q}_t) \\ \mathbf{P}_{t+1} &= N_t [1 - f_1(\mathbf{P}_t)] \\ \mathbf{Q}_{t+1} &= N_t f_1(\mathbf{P}_t) [1 - f_2(\mathbf{Q}_t)] \end{aligned}$$

In the above equations the subscripts t and $t + 1$ indicate two successive generations of \mathbf{P} , \mathbf{Q} and \mathbf{N} ; λ 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 \mathbf{P}_t or \mathbf{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 \mathbf{P} acts first, to be followed by \mathbf{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 \mathbf{P} and \mathbf{Q} act together on the same host stage, but the larvae of \mathbf{P} always out-compete those of \mathbf{Q} , should multi-parasitism occur.

The functions f_1 and f_2 are:

$$\begin{aligned} f_1(\mathbf{P}_t) &= \left[1 + \frac{a_1 \mathbf{P}_t}{k_1} \right]^{-k_1} \\ f_2(\mathbf{Q}_t) &= \left[1 + \frac{a_2 \mathbf{Q}_t}{k_2} \right]^{-k_2} \end{aligned}$$

where a_1 , a_2 , k_1 and k_2 are constants.

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

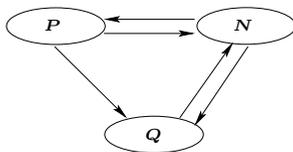


FIGURE 7. A sample biological network.

This gives

$$\begin{aligned} f_1(\mathbf{P}_t) &= 1 - a_1\mathbf{P}_t \\ f_2(\mathbf{Q}_t) &= 1 - a_2\mathbf{Q}_t \end{aligned}$$

and therefore, the equations are

$$\begin{aligned} (1, t): \mathbf{N}_{t+1} &= \lambda N_t(1 - a_1\mathbf{P}_t)(1 - a_2\mathbf{Q}_t) \\ (2, t): \mathbf{P}_{t+1} &= a_1 N_t \mathbf{P}_t \\ (3, t): \mathbf{Q}_{t+1} &= a_2 \mathbf{Q}_t N_t (1 - a_1\mathbf{P}_t) \end{aligned}$$

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

$$\mathbf{N} = \lambda \mathbf{N}(1 - a_1\mathbf{P})(1 - a_2\mathbf{Q}) \quad (16)$$

$$\mathbf{P} = a_1 \mathbf{N} \mathbf{P} \quad (17)$$

$$\mathbf{Q} = a_2 \mathbf{Q} \mathbf{N}(1 - a_1\mathbf{P}) \quad (18)$$

It can be easily seen from the above equations that one of the solutions is $\mathbf{P} = \mathbf{Q} = \mathbf{N} = 0$ (the “all zero” solution). If we ignore it, we get from (17) that

$$\mathbf{N} = \frac{1}{a_1} \quad (19)$$

and from (18) we get

$$1 = a_2 \cdot \frac{1}{a_1} (1 - a_1\mathbf{P}) \quad (20)$$

and hence

$$\begin{aligned} a_1 &= a_2 - a_2 a_1 \mathbf{P} \\ \mathbf{P} &= \frac{a_2 - a_1}{a_1 a_2} \end{aligned}$$

From (16), we get

$$1 = \lambda \left(1 - \frac{a_1(a_2 - a_1)}{a_1 a_2}\right) (1 - a_2\mathbf{Q})$$

$$1 = \frac{\lambda a_1}{a_2} (1 - a_2\mathbf{Q})$$

so

$$\frac{a_2}{\lambda a_1} = 1 - a_2\mathbf{Q}$$

$$a_2\mathbf{Q} = \frac{\lambda a_1 - a_2}{\lambda a_1}$$

$$\mathbf{Q} = \frac{\lambda a_1 - a_2}{\lambda a_1 a_2}$$

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

$$\begin{aligned}
(1) \quad 1 &= 2 \left(1 - 2 \cdot \frac{1}{6}\right) \left(1 - \frac{3}{12}\right) \\
&= 2 \cdot \frac{2}{3} \cdot \frac{9}{12} \\
&= 2 \cdot \frac{18}{36} \\
&= 1 \\
(2) \quad 1 &= 2 \cdot \frac{1}{2} \\
&= 1 \\
(3) \quad 1 &= 3 \cdot \frac{1}{2} \left(1 - \frac{2}{6}\right) \\
&= \frac{3}{2} \cdot \frac{4}{6} \\
&= 1
\end{aligned}$$

Let us substitute a_1 , a_2 and λ in the equations and pretend we do not know the solution. We get the equations:

$$\begin{aligned}
(1^*) \quad N &= 2N(1 - 2P)(1 - 3Q) \\
(2^*) \quad P &= 2PN \\
(3^*) \quad Q &= \frac{3}{2}Q(1 - 2P)
\end{aligned}$$

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

The equations above give rise to the iteration equations

$$\begin{aligned}
(1^*, i): \quad N_{i+1} &= 2N_i(1 - 2P_i)(1 - 3Q_i) \\
(2^*, i): \quad P_{i+1} &= 2N_iP_i \\
(3^*, i): \quad Q_{i+1} &= \frac{3}{2}Q_i(1 - 2P_i)
\end{aligned}$$

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'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

$$\begin{aligned}
N_1 &= 2 \cdot \frac{1}{2} \cdot N_i(1 - 1) \left(1 - \frac{3}{2}\right) = 0 \\
P_1 &= 2 \cdot \frac{1}{2} \cdot \frac{1}{2} = 0 \\
Q_1 &= \frac{3}{2} \cdot Q_i(1 - 2P_i) = 0 \\
N_2 &= 0 \\
P_2 &= 0 \\
Q_2 &= 0
\end{aligned}$$

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's method 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 schema, a schema 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. Iterating the equations will give us

$$\begin{aligned} N_1 &= 2 \\ P_1 &= 0 \\ Q_1 &= 0 \end{aligned}$$

$$\begin{aligned} N_k &= 2^k \\ P_k &= 0 \\ Q_k &= 0 \end{aligned}$$

This does not lead to a solution. It diverges!

The reader can check that even if the initial values are very close to a solution, the method in general will not converge to the solution.

Remark A.2. 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.3. Consider Figure 7 again but this time as an argumentation network where N , P , Q are arguments. This network has three extensions E_1 , E_2 and E_3 , namely

$$\begin{aligned} E_1 &= P \text{ is in} \\ &= N \text{ and } Q \text{ are out} \end{aligned}$$

$$\begin{aligned} E_2 &= N \text{ is in} \\ &= P \text{ and } Q \text{ are out} \end{aligned}$$

$$E_3 = P, N \text{ and } Q \text{ are all und}$$

In [13, 14, 15], we showed how to provide semantics for abstract argumentation in terms of equations. These equations are generated according to equation schema, of which two of the most significant ones are Eq_{max} and Eq_{inv} , described next.

Let $Att(X) = \{Y_1, \dots, Y_k\}$ be all the attackers of X . Consider X , Y_1, \dots, Y_k as variables ranging over $[0, 1]$. Define

$$\begin{aligned}
G_{max}(Att(X)) &= 1 - \max\{Y_1, \dots, Y_k\} \\
G_{inv}(Att(X)) &= \prod_{i=1}^k (1 - Y_i)
\end{aligned}$$

The equation we write for a node X is

$$X = G(Att(X)) \quad (*)$$

where G can be G_{max} or G_{inv} or some other function. We consider $X = 1$ to mean X is **in**; $X = 0$ to mean X is **out**; and $0 < X < 1$ to mean that X is **und**. The background material on the equational approach is given in the next section. It is sufficient to say here that G_{max} follows more closely the traditional semantics of argumentation networks being only concerned about the highest strength of attack to a node. The solutions to the equations using G_{max} correspond to the traditional concept of extensions (in Dung's sense) taking the nodes with value 1 in a solution to be the nodes in the extension.

G_{inv} on the other hand is also sensitive to the number of attackers to a node. For example, assume there are 10 undecided attackers Y_i of X each having value $\frac{1}{2}$ (**und**), then the value of X becomes $\frac{1}{2^{10}}$ under G_{inv} , while under G_{max} , the value of X is simply $\frac{1}{2}$. Note that X is nearer to 0 (i.e., **out**) in the G_{inv} case!

The G_{max} equations for the network in Figure 7 are:

$$N = 1 - \max\{P, Q\} \quad (21)$$

$$P = 1 - N \quad (22)$$

$$Q = 1 - \max\{P, N\} \quad (23)$$

and its G_{inv} equations are:

$$N = (1 - P)(1 - Q) \quad (24)$$

$$P = (1 - N) \quad (25)$$

$$Q = (1 - P)(1 - N) \quad (26)$$

The G_{max} equations have the solutions: $N = Q = 0$ and $P = 1$ (E_1); $N = 1, P = Q = 0$ (E_2); and $N = P = Q = \frac{1}{2}$ (E_3). The G_{inv} only accepts the first two solutions with the extension E_3 not being possible.²⁰

Now suppose we actually do not know whether there are solutions or what they would be and let us consider our options. We have a system of equations involving N, P and Q and we want to try and solve it.

Option 1 – A mathematical view. Let us just find a solution. This is a numerical analysis problem. We can guess a candidate solution; use, for instance, Newton's method; and iterate in the hope of converging to a solution. **Option**

2 – A semantical view. We seek a solution motivated not by mathematics

²⁰The specific behaviour of G_{inv} is outside of the scope of this paper. However it is explored in detail in [12].

but by the meaning of the equations; by argumentation considerations. Newton's method may not be adequate here. We want a method which, if we start very near a solution, then we get convergence to that desired solution. Here we cannot accept any solution. We want solutions which reflect the input. So we need to devise algorithms involving iterations which have a semantic meaning, in addition to the usual mathematical properties that the iteration sequences calculated by these algorithms converge. This point is important. Suppose we give the following interpretation to the network. 100 voters need to form a committee from amongst three experts \mathbf{P} , \mathbf{Q} and \mathbf{N} to give an opinion on a crucial issue. All of them vote for \mathbf{N} to be included (**in**), none of them want \mathbf{P} to be included (i.e., they want \mathbf{P} to be **out**), and they are equally divided on their support for \mathbf{Q} (**und**). There is however an additional information about these candidates which is of a personal nature of which the voters are not aware. These are represented by the attack relation in the network, in which $X \rightarrow Y$ means X refuses to work with Y . We thus say that we have a numerical assignment $\mathbf{N} = 1$, $\mathbf{P} = 0$ and $\mathbf{Q} = \frac{1}{2}$ and we now ask what extension (i.e., what committee membership) is nearest to this majority vote? At first glance, the reader may think that it is extension E_2 (\mathbf{N} is **in**, and \mathbf{P} and \mathbf{Q} are **out**), because it agrees with the wishes of all of the voters that \mathbf{N} is **in** and \mathbf{P} is **out**. We would like our iteration algorithm to give us this result if possible.

Let us look at what Newton's method would do to these initial values.

We start with initial values $\mathbf{N}_0 = 1$, $\mathbf{P}_0 = 0$ and $\mathbf{Q}_0 = \frac{1}{2}$ and iterate for the case of G_{max} (equations (21)–(23)). We shall see that iterating in this way is not satisfactory. We get

$$\begin{aligned} \mathbf{N}_1 &= \frac{1}{2}, & \mathbf{P}_1 &= 0, & \mathbf{Q}_1 &= 0 \\ \mathbf{N}_2 &= 1, & \mathbf{P}_2 &= \frac{1}{2}, & \mathbf{Q}_2 &= \frac{1}{2} \\ \mathbf{N}_3 &= \frac{1}{2}, & \mathbf{P}_3 &= 0, & \mathbf{Q}_3 &= 0 \end{aligned}$$

There is no convergence here, so this is not satisfactory as we do not get an answer for membership (i.e., no extension in the argumentation sense).

Let us now compare with the Gabbay-Rodrigues Iteration Schema for G_{max} , which is the main subject matter of this paper and is introduced in Section 2. The schema always yields a solution which corresponds to an extension in the argumentation sense.

Let $\langle S, R \rangle$ be an argumentation network and $X, Y_i \in S$ be considered variables. Let $Att(X) = \{Y_j\}$ ($j \geq 0$) be the attackers of X and let the equations be $X = G_{max}(Att(X))$.²¹ Let $V_i(X)$ be the value of X at iteration step i . Then the value of X at step $i + 1$ is calculated as

$$\begin{aligned} V_{i+1}(X) &= (1 - V_i(X)) \cdot \min \left\{ \frac{1}{2}, G(\{V_i(Y_j)\}) \right\} + \\ &V_i(X) \cdot \max \left\{ \frac{1}{2}, G(\{V_i(Y_j)\}) \right\} \end{aligned}$$

²¹ G_{inv} can also be used, with different results.

So for the network in Figure 7 and G_{max} we get

$$\begin{aligned}
V_{i+1}(\mathbf{N}) &= (1 - V_i(\mathbf{N})) \cdot \min \left\{ \frac{1}{2}, 1 - \max\{V_i(\mathbf{P}), V_i(\mathbf{Q})\} \right\} + \\
&\quad V_i(\mathbf{N}) \cdot \max \left\{ \frac{1}{2}, 1 - \max\{V_i(\mathbf{P}), V_i(\mathbf{Q})\} \right\} \\
V_{i+1}(\mathbf{P}) &= (1 - V_i(\mathbf{P})) \cdot \min \left\{ \frac{1}{2}, 1 - V_i(\mathbf{N}) \right\} + \\
&\quad V_i(\mathbf{P}) \cdot \max \left\{ \frac{1}{2}, 1 - V_i(\mathbf{N}) \right\} \\
V_{i+1}(\mathbf{Q}) &= (1 - V_i(\mathbf{Q})) \cdot \min \left\{ \frac{1}{2}, 1 - \max\{V_i(\mathbf{P}), V_i(\mathbf{N})\} \right\} + \\
&\quad V_i(\mathbf{Q}) \cdot \max \left\{ \frac{1}{2}, 1 - \max\{V_i(\mathbf{P}), V_i(\mathbf{N})\} \right\}
\end{aligned}$$

Let us now take the initial conditions $V_0(\mathbf{N}) = 1$, $V_0(\mathbf{P}) = 0$ and $V_0(\mathbf{Q}) = 0$ and calculate the iterations. All values will converge to $\frac{1}{2}$.

The perceptive reader might ask what is the philosophy behind the schema that led us to the extension E_3 , rather than to the larger extension E_2 . The schema is very sensitive to the undecided values. It acts cautiously in considering the votes for \mathbf{N} 's being included, because a proportion of the voters wanted \mathbf{Q} to be included but \mathbf{N} and \mathbf{Q} cannot work together.

Appendix 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 [17], 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;
- Π 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 $h(V_0(X), g_{Y \in Att(X)}(\{1 - V_e(Y)\}))$ and computed by the algorithm II. Since the computation of the equilibrium values of the nodes takes the values of the attacking nodes into account, in Cayrol and Lagasque-Schieux's terminology, the algorithm II offers a procedure to perform an *interaction-based valuation* of the graph $\langle S, R \rangle$. 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 8 to illustrate how numerical argumentation networks are used in the context of the argumentation-friendly functions seen in this paper.

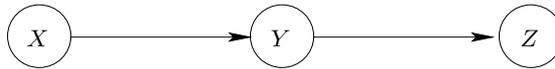


FIGURE 8. A simple argumentation graph without cycles.

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 9.

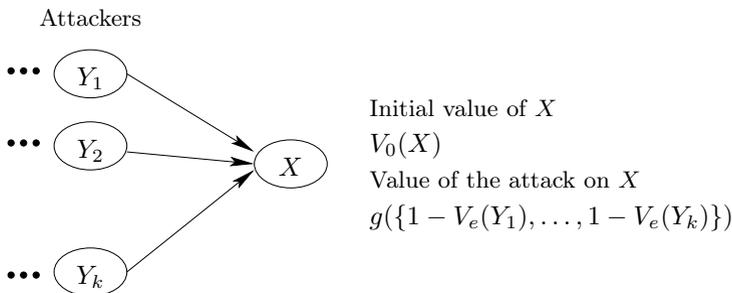


FIGURE 9. Attacks to a node and its initial value.

We can compute $g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\}) = y$, 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 10.

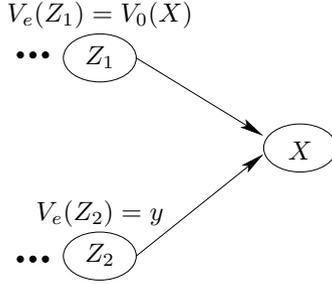


FIGURE 10. Combination of a node's initial value with its attacks.

and compute $h(V_e(Z_1), V_e(Z_2))$, i.e., $h(V_0(X), g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\}))$. We get equations of the kind

$$V_e(X) = h(V_0(X), g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\})) \quad (27)$$

to solve. As we mentioned, g and h may be different functions, so for example we could have $g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\}) = \min(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\})$ 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 11. And then we get $V_e(X) = \min(\{1 - (1 - V_0(X)), 1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\}) = \min(\{V_0(X), 1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\})$. 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 $V_e(X) = h(1, g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\})) = g(\{1 - V_e(Y_1), \dots, 1 - V_e(Y_k)\})$.

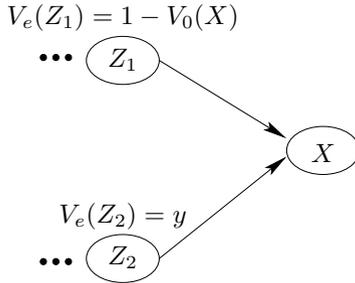


FIGURE 11. Combining attacks and initial value.

We now address another issue. Once we solve equation (27), we get a function V_e such that

$$V_e(X) = h(V_0(X), g(\{1 - Y_1, \dots, 1 - Y_k\}))$$

Can we use $V_e(X)$ itself as an initial value?

In other words, do we have that equation (28) below holds?

$$V_e(X) = h(V_e(X), g(\{1 - Y_1, \dots, 1 - Y_k\})) \quad (28)$$

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

$$\begin{aligned} V_e(X) &= h(V_e(X), g(\{1 - Y_1, \dots, 1 - Y_k\})) \\ &= g(\{V_e(X), g(\{1 - Y_1, \dots, 1 - Y_k\})\}) \\ &= g(\{V_e(X), 1 - Y_1, \dots, 1 - Y_k\}) \\ &= g(\{Z, 1 - Y_1, \dots, 1 - Y_k\}) \end{aligned}$$

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_X^1 and Z_X^2 for each X and form the graph depicted in Figure 12. All solutions to the cycle

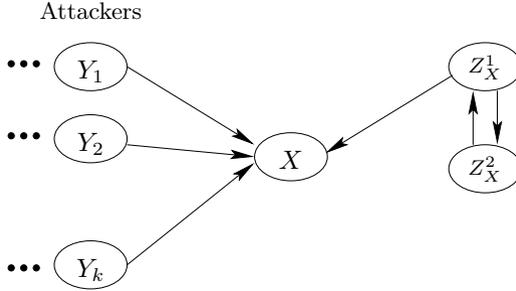


FIGURE 12. Combining attacks and initial value.

$Z_X^1 \leftrightarrow Z_X^2$ are of the form $(V_e(Z_X^1), 1 - V_e(Z_X^1))$, which means that Z_X^1 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

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

giving us our now familiar Eq_{\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*.

$$\prod_{Y \in Att(X)} (1 - V_e(Y)) \quad (29)$$

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 (29) is equivalent to

$$1 - \gamma_{Y \in Att(X)} V_e(Y) \tag{30}$$

where $x \gamma y = x + y - x.y$ and for $V = \{x_1, \dots, x_k\}$, $\gamma V = (((x_1 \gamma x_2) \gamma \dots) \gamma x_k)$. (30) 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.

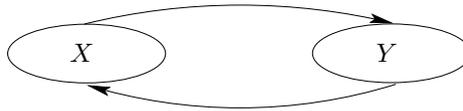


FIGURE 13. A cycle involving two nodes.

Assume that all initial values are 1, that g and h are product. The graph in Figure 13 will generate the system of equations

$$\begin{aligned} V_e(X) &= 1 - V_e(Y) \\ V_e(Y) &= 1 - V_e(X) \end{aligned}$$

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 $\kappa \rightarrow 1$. This would give us

$$\begin{aligned}
V_e(X) &= \kappa(1 - V_e(Y)) \\
V_e(Y) &= \kappa(1 - V_e(X)) \\
\\
V_e(X) &= \kappa - \kappa V_e(Y) \\
&= \kappa - \kappa(\kappa - \kappa V_e(X)) \\
&= \kappa - \kappa^2 + \kappa^2 V_e(X) \\
V_e(X) - \kappa^2 V_e(X) &= \kappa - \kappa^2 \\
V_e(X)(1 - \kappa^2) &= \kappa - \kappa^2 \\
V_e(X) &= \frac{\kappa(1 - \kappa)}{(1 - \kappa)(1 + \kappa)} \\
V_e(X) &= \frac{\kappa}{1 + \kappa}
\end{aligned}$$

Hence, when $\kappa \rightarrow 1$, $V_e(X) = V_e(Y) = \frac{1}{2}$. This result explains the implicit introduction of the parameter ε to the vote aggregation function proposed by Leite and Martins in [19].²²

Since the initial values of the two nodes in the network of Figure 13 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 14.

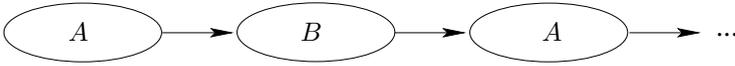


FIGURE 14. Unravelling the cycle in the network of Figure 13.

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

$$V_e(X) = x \cdot (1 - (x \cdot (1 - (x \cdot (1 - \dots))))))$$

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

$$V_e(X) = \frac{1}{1+\varepsilon} \left(1 - \left(\frac{1}{1+\varepsilon} \left(1 - \left(\frac{1}{1+\varepsilon} (1 - \dots) \right) \right) \right) \right)$$

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

$$\delta = 1 - \left(\frac{1}{1+\varepsilon} \left(1 - \left(\frac{1}{1+\varepsilon} (1 - \dots) \right) \right) \right)$$

²²We 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 4.

Let us calculate what the value δ is. To simplify the calculation we set $\alpha = (1 + \varepsilon)$, we then get

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

If we expand the first multiplication, we get

$$\begin{aligned} \delta &= 1 - \left(\frac{1}{\alpha} - \frac{1}{\alpha^2} \left(1 - \frac{1}{\alpha} (\dots) \right) \right) \\ &= 1 - \left[\frac{1}{\alpha} - \frac{1}{\alpha^2} + \frac{1}{\alpha^3} \left(1 - \frac{1}{\alpha} (\dots) \right) \right] \\ &= 1 - \left[\frac{1}{\alpha} - \frac{1}{\alpha^2} + \frac{1}{\alpha^3} - \frac{1}{\alpha^4} \left(1 - \frac{1}{\alpha} (\dots) \right) \right] \\ &= 1 - \left[\left(\frac{\alpha - 1}{\alpha^2} \right) + \left(\frac{\alpha - 1}{\alpha^4} \right) + \left(\frac{\alpha - 1}{\alpha^6} \right) + \dots \right] \end{aligned}$$

The component

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

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

$$\begin{aligned} & \frac{\alpha^2(\alpha - 1)}{\alpha^2 - 1} - (\alpha - 1) \\ &= \frac{\alpha^2(\alpha - 1) - (\alpha^2 - 1)(\alpha - 1)}{\alpha^2 - 1} \\ &= \frac{(\alpha - 1)(\alpha^2 - \alpha^2 + 1)}{\alpha^2 - 1} \\ &= \frac{\alpha}{\alpha^2 - 1} \end{aligned}$$

Remember that $\alpha = 1 + \varepsilon$, hence

$$\begin{aligned} \frac{\alpha}{\alpha^2 - 1} &= \frac{1 + \varepsilon - 1}{(1 + \varepsilon)(1 + \varepsilon) - 1} \\ &= \frac{\varepsilon}{\varepsilon^2 + 2\varepsilon + 1 - 1} \\ &= \frac{\varepsilon}{\varepsilon(\varepsilon + 2)} \\ &= \frac{1}{\varepsilon + 2} \end{aligned}$$

Therefore,

$$\delta = \left(1 - \frac{1}{\varepsilon + 2}\right)$$

and hence in the limit $\varepsilon \rightarrow 0$, we get

$$V_e(X) = \lim_{\varepsilon \rightarrow 0} \frac{1}{1 + \varepsilon} \left(1 - \frac{1}{\varepsilon + 2}\right) = \frac{1}{2}$$

as expected.

If we just have an acyclic sequence of attacks such as the one in Figure 15, 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).



FIGURE 15. Sequence of attacks.

From the network in Figure 15, 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 $V_e(X_1) = 1$, $V_e(X_2) = 0$, $V_e(X_3) = 1, \dots$. 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 $V_e(X_1) = \frac{1}{2}$, $V_e(X_2) = \frac{1}{4}$, $V_e(X_3) = \frac{3}{8}$, \dots

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

$$h(X) = \begin{cases} 1, & \text{if } Att(X) = \emptyset \\ \frac{1}{1 + \sum_{Y \in Att(X)} h(Y)}, & \text{otherwise} \end{cases}$$

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 16.



FIGURE 16. Attack by a node of same initial value.

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.²³ 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 17.

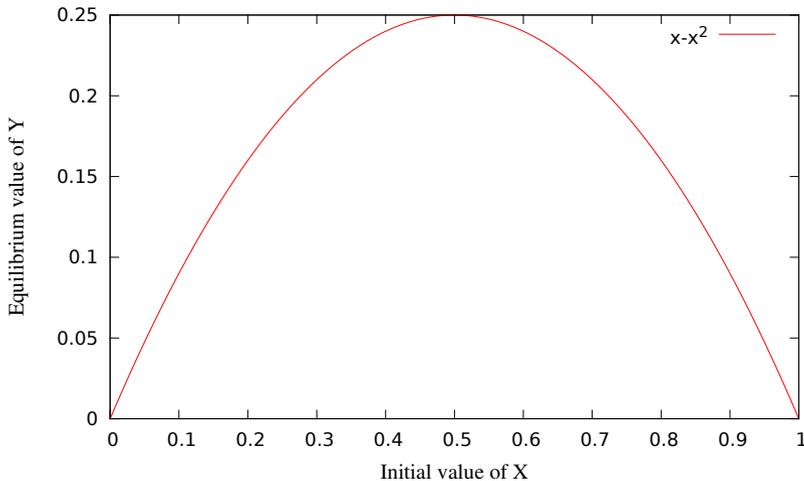


FIGURE 17. Attack by a single node of same initial value.

²³This 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 [19], 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 Eğılmez *et al.*’s work [11], which is an extension to [19] so we can compare it with our methodology.²⁴

Definition B.2. [11] *A social abstract argumentation framework is a tuple $\langle \mathcal{S}, \mathcal{R}, V_S, V_R \rangle$, where \mathcal{S} is a set of arguments; $\mathcal{R} : \mathcal{S} \times \mathcal{S}$ is a binary attack relation between arguments; and $V_S : \mathcal{S} \rightarrow \mathbb{N} \times \mathbb{N}$ and $V_R : \mathcal{R} \rightarrow \mathbb{N} \times \mathbb{N}$ 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, Eğılmez *et al.* introduce the concept of a *semantic framework* presented below.

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

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

The operations τ , λ , Υ 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 $[0, 1]$); λ_S and λ_R are both the

²⁴Note that [19] were not aware (and did not quote) [1], which was six years earlier. Thus, the only new contribution in [1] was how they determine the initial values and the connection with voting.

product t-norm \wedge , where $x \wedge y = x.y$; Υ is its associated t-conorm, i.e., $x \Upsilon y = 1 - (1 - x).(1 - y) = x + y - x.y$; $\neg x = 1 - x$; and τ is one of a family of operations τ_ε 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 τ_ε 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 ε is).²⁵

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 $\mathcal{T} = \langle L, \tau, \wedge_S, \wedge_R, \Upsilon, \neg \rangle$ a semantic framework. A social model of F under semantics \mathcal{T} is a total mapping $M : \mathcal{S} \rightarrow L$ such that for every $X \in \mathcal{S}$

$$M(X) = \tau(X) \wedge \neg \Upsilon_{Y_i \in Att(X)} \{ \tau((Y_i, X)) \wedge M(Y_i) \}$$

Note that if \wedge is product t-norm and Υ is its t-conorm, as in [11], then

$$\begin{aligned} M(X) &= \tau(X) \wedge \neg \Upsilon_{Y_i \in Att(X)} \{ \tau((Y_i, X)) \wedge M(Y_i) \} \\ &= \tau(X) \cdot \left(1 - \left(1 - \prod_{Y_i \in Att(X)} (1 - \tau((Y_i, X)) \cdot M(Y_i)) \right) \right) \\ &= \tau(X) \cdot \prod_{Y_i \in Att(X)} (1 - \tau((Y_i, X)) \cdot M(Y_i)) \end{aligned}$$

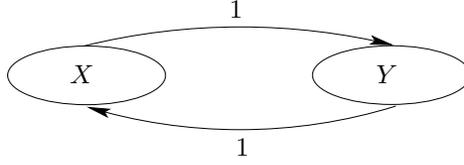
Contrast $M(X)$ with the equilibrium value of X , $V_e(X)$ as we proposed it in [17, Definition 5]:

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

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 [19], there are difficulties with the vote aggregation function τ . At first, the constant ε was introduced to avoid the existence of infinite models. For example, consider the network

²⁵ ε cannot be 0, because this would render τ_ε ill defined for components with no votes.



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 ε that is nevertheless greater than 0, we get the following situation.

$$\begin{aligned} M(X) &= \frac{1}{1 + \varepsilon}(1 - M(Y)) \\ M(Y) &= \frac{1}{1 + \varepsilon}(1 - M(X)) \end{aligned}$$

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

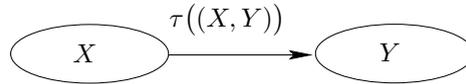
$$\begin{aligned} M(X) &= \frac{1}{1 + \varepsilon} \left(1 - \frac{1}{1 + \varepsilon}(1 - M(X)) \right) \\ &= \frac{1}{1 + \varepsilon} \left(\frac{1 + \varepsilon - 1 + M(X)}{1 + \varepsilon} \right) \\ &= \frac{1}{1 + \varepsilon} \left(\frac{\varepsilon + M(X)}{1 + \varepsilon} \right) \\ &= \frac{\varepsilon + M(X)}{(1 + \varepsilon)^2} \\ M(X)(1 + \varepsilon)^2 &= \varepsilon + M(X) \\ M(X)(1 + \varepsilon)^2 - M(X) &= \varepsilon \\ M(X) &= \frac{\varepsilon}{(1 + \varepsilon)^2 - 1} \\ &= \frac{\varepsilon}{2\varepsilon + \varepsilon^2} \\ &= \frac{1}{2 + \varepsilon} \end{aligned}$$

and hence $\lim_{\varepsilon \rightarrow 0} M(X) = \frac{1}{2} = M(Y)$, 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 τ is to calculate initial support for components and in that respect, the constant ε has no part to play. This situation does not arise in [17, 16], 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:



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.²⁶ 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.²⁷

D. M. Gabbay
 King's College London,
 Department of Informatics,
 The Strand,
 London, WC2R 2LS, UK, and
 Bar Ilan University, Israel, and
 University of Luxembourg
 e-mail: `dov.gabbay@kcl.ac.uk`

O. Rodrigues
 King's College London,
 Department of Informatics,
 The Strand,
 London, WC2R 2LS, UK

e-mail: `odinaldo.rodrigues@kcl.ac.uk`

²⁶The main motivation for the introduction of the weights on attacks in [11].

²⁷High values of τ should correspond to high level of initial support.