Argument and Computation
Vol. 3, Nos. 2–3, June–September 2012, 143–202

Temporal, numerical and meta-level dynamics in argumentation networks

H. Barringer[a], D.M. Gabbay[b,c,d]* and J. Woods[e]

[a]*Department of Computer Science, University of Manchester, Manchester, M13 9PL, UK;* [b]*Department of Computer Science, Bar-Ilan University, 52900 Ramat-Gan, Israel;* [c]*Department of Informatics Science, King's College London, Strand, London WC2R 2LS, UK;* [d]*Computer Science and Communications, Faculty of Sciences, 6, rue Coudenhove-Kalergi, L-1359, Luxembourg;* [e]*Department of Philosophy, Univeristy British Columbia, Vancouver, Canada*

(*Received 11 April 2011; final version received 15 June 2012*)

This paper studies general numerical networks with support and attack. Our starting point is argumentation networks with the Caminada labelling of three values $1 =$ in, $0 =$ out and $\frac{1}{2} =$ undecided. This is generalised to arbitrary values in [01], which enables us to compare with other numerical networks such as predator–prey ecological networks, flow networks, logical modal networks and more. This new point of view allows us to see the place of argumentation networks in the overall landscape of networks and import and export ideas to and from argumentation networks. We make a special effort to make clear how general concepts in general networks relate to the special case of argumentation networks. We pay special attention to the handling of loops and to the special features of numerical support. We find surprising connections with the Dempster–Shafer rule and with the cross-ratio in projective geometry. This paper is an expansion of our 2005 paper and so we also consider higher level features such as numerical attacks on attacks, and propagation of numerical values.We conclude with a brief view of temporal numerical argumentation and with a detailed comparison with related papers published since 2005.

Keywords: argumentation network; numerical networks; higher level attacks; strength of arguments; analysis of support

1. Introduction and orientation

This paper is an expansion of our 2005 Festschrift paper (Barringer, Gabbay, andWoods 2005). The argumentation community was not aware, until recently, of Barringer et al. (2005) and some of the ideas there were rediscovered in a related form by various people at a later date, see for example papers (Cayrol and Lagasquie-Schiex 2005; Martinez, Garcia, and Simari 2008; Matt and Toni 2008; Modgil and Bench-Capon 2008; Baroni, Cerutti, Giacomin, and Guida 2009a; Boella, Kaci, van der Torre 2009; Dunne, Hunter, McBurney, Parsons, and Wooldridge 2009; Haenni 2009; Cayrol, Devred, and Lagasquie-Schiex 2010; Cobo, Martinez, and Ricardo Simari 2010; Modgil and Bench-Capon 2010; Rotstein, Moguillansky, Javier Garcia, and Ricardo Simari 2010; Baroni, Cerutti, Giacomin, and Guida 2011; Dunne, Hunter, McBurney, Parsons, and Wooldridge 2011; Leite and Martins 2011; Gabbay and Rodrigues 2012). However, Barringer et al. (2005) remains the most general approach, containing ideas still new to the community (see also Gabbay 2009a). It is high time to make an expanded journal paper of this material.

In the first section, we provide a landscape orientation and an introduction to give the reader a perspective, and in the last section, we offer a comparison with the literature.

Our starting point is a network (S, R), where S is a set of arguments and $R \subseteq S \times S$ is an attack relation between arguments.

*Corresponding author. Email: dov.gabbay@kcl.ac.uk

ISSN 1946-2166 print/ISSN 1946-2174 online

http://dx.doi.org/10.1080/19462166.2012.704397
http://www.tandfonline.com

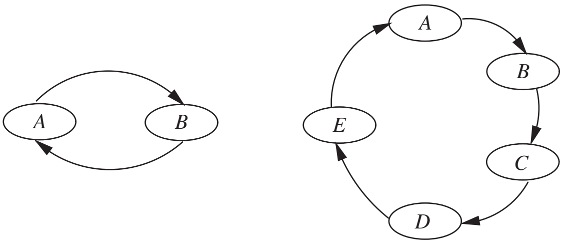


Figure 1. Sample argumentation networks.

Figure 1 gives examples of two such networks.

The networks in Figure 1 are in fact loops involving even and odd number of arguments. An arrow $x \longrightarrow y$ indicates an attack from node x to node y, i.e. that $(x, y) \in R$. There are three ways of looking at such networks.

(1) Dung's (1995) original approach, where properties of subsets $E \subseteq S$ are considered and a set-theoretical definition of *extension* is put forward.
The basic notion is that of admissible extension. This is a subset $E \subseteq S$ such that
(a) E is conflict-free, i.e. for no $x, y \in E$ can we have $(x, y) \in R$.
(b) E is self-defending, i.e. for all $x, y \in S$, if $x \in E$ and $(y, x) \in R$ then there exists a $z \in S$ such that $(z, y) \in R$.
In the network on the left of Figure 1, there are two extensions $E_1 = \{A\}$ and $E_2 = \{B\}$ as well as the empty extension $E_0 = \varnothing$. The network on the right of the figure has only the empty extension.

(2) Recently, Caminada (see Caminada (2006) and Caminada and Gabbay (2009) for a survey) came forward with the brilliant idea of a Caminada labelling function $\lambda : \mathcal{S} \longrightarrow \{\text{in}, \text{out}, \text{undecided}\}$, satisfying certain conditions that ensure a complete correspondence with Dung's extensions (Caminada and Gabbay 2009; Caminada 2011). For example, for the network on the left of Figure 1, we have the three functions λ_1, λ_2 and λ_0 below.

E_1	E_2	E_0
$\lambda_1(A) = \text{in}$	$\lambda_2(A) = \text{out}$	$\lambda_0(A) = \text{undecided}$
$\lambda_1(B) = \text{out}$	$\lambda_2(B) = \text{in}$	$\lambda_0(B) = \text{undecided}$

For the network on the right of Figure 1, we have only the function λ such that $\lambda(A) = \lambda(B) = \lambda(C) = \lambda(D) = \lambda(E) = \text{undecided}$.

(3) The third approach is Gabbay's (2011a, c) and (2012a, b), equational approach. This approach views (S, R) as a mathematical graph generating equations for functions in the unit interval $U = [0, 1]$. Any solution $\mathbf{f}$ 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 are viewed as the totality of the equational extensions. Given the situation described in Figure 2, where $A \in S$ is any node and $X_1, \ldots, X_n$ are all of the nodes attacking A (i.e. $(X_i, A) \in R$), one equation we can possibly generate is Eq_{max}:

$$\mathbf{f}(A) = 1 - \max_i\{\mathbf{f}(X_i)\}. \qquad (\text{Eq}_{\text{max}})$$

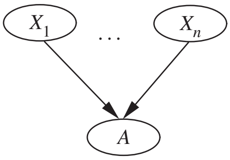


Figure 2. Direct attacks on node A.

Another possibility is $\mathrm{Eq}_{\mathrm{inv}}$:

$$\mathbf{f}(A) = \prod_{i=1}^{n}(1 - \mathbf{f}(X_i)). \tag{$\mathrm{Eq}_{\mathrm{inv}}$}$$

Gabbay has shown that in the case of $\mathrm{Eq}_{\max}$ the totality of solutions corresponds to the totality of extensions in Dung's sense. The correspondence is best explained in terms of the Caminada labelling. We get the Caminada labelling through the correspondence below.

$$\begin{aligned}
\mathbf{f}(A) = 1 &\quad :: \lambda(A) = \text{in},\\
\mathbf{f}(A) = 0 &\quad :: \lambda(A) = \text{out},\\
0 < \mathbf{f}(A) < 1 &\quad :: \lambda(A) = \text{undecided}.
\end{aligned}$$

The above discussion relates to abstract argumentation networks, where (S, R) is a pure graph. The nodes of the graph, in this case, have no internal structure and the relation $(x, y) \in R$ is given without any further structural explanation.

Caminada and other colleagues hold the view that such networks are too abstract and further structure is needed for practical applications. Caminada recommends a base logic $\boldsymbol{L}$ on some language and a base theory Δ. The arguments $A \in S$ are derived from proofs in $\boldsymbol{L}$ from Δ. Thus, A attacks B happens now for a reason: the contents of A attack (logically in $\boldsymbol{L}$) the contents of B.

Another enrichment to purely abstract argumentation networks is to add a preference relation on S to tell us which abstract arguments are preferred to others.

A third alternative is obtained by adding a valuation $V : S \longrightarrow$ VALUES, where VALUES is a set of abstract values, to give us a little more information about the elements of S. Mathematically one can get a preference relation out of a valuation and a valuation out of a preference relation. So let us view our position mathematically as being given a triple (V, S, R) where (S, R) is an abstract argumentation network and V a valuation function as before.

Given (V, S, R), we now have two main options.

(1) to involve V in the definition of extensions
(2) to define the extensions without using V, but involve it in the choice of extensions once these are generated. For example, if the valuation give us that A has the highest value in network R of Figure 1, we may adopt the new extension $\{A\}$.

Option 1 was widely used by Bench–Capon and colleagues, while option 2 is favoured by Caminada and his colleagues, as well as by Talmudic Logic (Abraham, Gabbay, and Schild 2011a,b).

In the equational approach, it is reasonable to assume that V is a function $V : S \cup R \longrightarrow U$ and thus option 1 means that we use V in formulating the equations, whereas option 2 means we use V to eliminate or modify some solutions $\mathbf{f}$. For example, going back to Figure 2, we now also have the values $V(A)$, $V(X_1), \dots, V(X_n)$ and so we can either modify the equations and write, for instance, $\mathrm{Eq}_{\mathrm{inv}}$ as

$$\mathbf{f}(A) = V(A) \cdot \prod_{i=1}^{n} (1 - \mathbf{f}(X_i))$$

or stick to the original equations, but accept only solution functions **f** such that, for example, for all $A \in \mathcal{S}$, $\mathbf{f}(A) \geq V(A)$.

The approach, we adopt in this paper, is numerical, that is, we look at argumentation networks (S, R, V) where $V : S \cup R \mapsto [0, 1]$. In 2005, in our paper (Barringer et al. 2005), this was a pretty innovative approach. In this framework, we also put forward new concepts, like attack on attacks to any level, see Section 2 below, an idea which applies to any network (not necessarily numerical) and was later rediscovered by Modgil (2009) and by Baroni et al. (2009a).

A full discussion of the subject matter of higher level attacks, including priority can be found in Gabbay (2009a).[1]

The study of this topic, in its present generality, has emerged from our previous research into argumentation frameworks (Gabbay and Woods 2001a,b, 2002, 2003a,b; Woods 2004), where we observed that circular loops in argumentation networks, no matter whether they are even or odd, can be viewed as, and possibly resolved as, local predator–prey networks. Our starting point is, therefore, a generalisation of the abstract argumentation networks.

Let us now consider some examples.

The numerical view of networks allows us to connect argumentation networks with other major networks.

Example 1.1 Figures 3–5 are three examples of such networks. We assume that the arguments denoted by the nodes all have equal and unit strength, and similarly all attacks have unit strength. The result of a source node attacking a target (all of unit strength) is the refutation of the target. Evaluating the effects of such a network will determine which arguments survive, i.e. are *active*, and which ones are refuted, i.e. are *inactive*.

(a) The situation in Figure 3 is straightforward. The argument a is not attacked by anything, so it is evaluated as an active argument. Since a attacks b, b is a refuted argument and is evaluated as inactive, and so, as c is not attacked by an active argument, c is evaluated as active. We can write the net, evaluated, result of Figure 3 as $\{+a, -b, +c\}$.

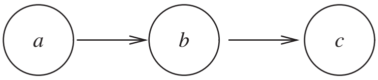


Figure 3. Example of network.

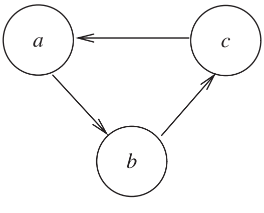


Figure 4. Example of network.

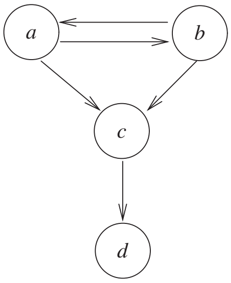


Figure 5. Example of network.

(b) The situation in Figure 4 is a complete loop. No argument can definitely be said to be active or inactive, and hence all three arguments denoted by a, b and c are evaluated as *unknown*. We write this as $\{?a, ?b, ?c\}$.

(c) The situation in Figure 5 is more interesting. Here, as a and b attack each other, we have those arguments evaluated as unknown, i.e. $\{?a, ?b\}$. Because of that, we also evaluate arguments c and d as unknown. If a or b were active, c would be refuted and evaluated as inactive and thus d would be evaluated as active. On the other hand, if neither a nor b were active, then c would evaluate as active, which in turn would cause d's evaluation to be inactive, (note that this does not give us an extension in the Dung–Caminada sense). However we can observe that both a and b attack c; so no matter which of a or b are active (i.e. whether we have $\{+a, -b\}$ or $\{-a, +b\}$), we always have $-c$, and so the net result could be taken to be $\{?a, ?b, -c, +d\}C$, (again note that this does not give us an extension in the Dung–Caminada sense). On the other hand, we might adopt the view that a, b cancel each other, in which case the net result would be $\{-a, -b, +c, -d\}$ (again note that this does not give us an extension in the Dung–Caminada sense).

Since circularity, loops and mutual attacks of arguments are very common in real life, it is obvious that much attention is required to resolving loops in argumentation networks. Abstract argumentation networks were generalised by Bench-Capon (2003), where a colouring (representing the type of argument) was added to the network. The colours are linearly ordered by strength. A weaker coloured node cannot successfully attack a stronger coloured node. So a network with colours has the form (S, R, V) where, as before, $R \subseteq S \times S$ and V is a function giving, say, numbers to nodes: $V : S \mapsto \ \mathit{Numbers}$, and the numbers represent strength.

Thus, in Figure 4, suppose $V(b) = r$ and $V(a) = V(c) = s$. Clearly if $r < s$, then the attack of b on c cannot take place and the net outcome of the network is $\{+b, +c, -a\}$. If $r > s$, then the attack of a on b cannot take place and the result is $\{+b, +a, -c\}$. If $r = s$, we get, as before, $\{?a, ?b, ?c\}$.

Note that technically the colouring function V is an instrument for cancelling attacks from some nodes to others. However, it is an instrument that requires restrictions. Not every proposed list of attacks to be cancelled can be implemented by a function V. Consider Figure 4. Suppose that we want to cancel all attacks. To cancel the attacks of a on b and of b on c we must have $V(a) < V(b) < V(c)$. By transitivity $V(a) < V(c)$, so the attack by c on a cannot be cancelled by V.

The main rationale behind the introduction of V is not necessarily the resolution of loops or cancellation of attacks, but the modelling of the intuition that arguments can be divided into kinds, and that some kinds of arguments are more important than others, some kinds of arguments are irrelevant,[2] etc.

Remark 1.2 As we mentioned earlier, this paper, following its earlier 2005 version (Barringer et al. 2005), goes in the numerical direction. This enables us to connect argumentation networks with a diverse landscape of other very well known networks and the communities studying them. We note that following the idea of Caminada labelling, the traditional Dung approach to argumentation networks can also be considered as numerical. For a node x, we give the values $V(x) = 1$ for x being in, $V(x) = 0$ for x being out and $V(x) = \frac{1}{2}$ for x being undecided. If $x_1, \ldots, x_n$ are all the attackers of y, then we let $V(y) = 1 - \max V(x_1), \ldots, V(x_n)$.

It is shown in Gabbay (2011a, c), that this particular three valued numerical approach is equivalent to the traditional Dung approach. In the sequel, we shall always point out how the new ideas and definitions of this paper manifest themselves in traditional Dung argumentation, by using the above special three valued numerical case.

The numerical view of networks allows us to connect argumentation networks with other major networks.

Let us list a few major ones.

(A) For Argumentation theory, node b may represent an argument or statement and the supporting or attacking nodes a_i would be arguments in favour or against the argument represented by node b. The associated weights on the nodes and arrows would indicate the strength of each argument, the strength of each support or attack and the force of their presentation. The main problem for a given network of this type is to determine how to evaluate the effects of the various supports and attacks in an untimed context with fixed weights as well as a timed context in which weights are treated as being time-dependent (see Caminada and Gabbay 2009; Gabbay and d'Avila Garcez 2009; Gabbay and Szalas 2009).

(B) The well-known Bayesian networks fall within our abstraction. Dependent nodes, i.e. target nodes, can be viewed as being evaluated through joint conditional probability on their parent nodes, i.e. source nodes. There are no individual weights on arrows unless there is independence of the conditional probabilities of the target on its sources.

(F) In Flow networks, where directed edges may be labelled by various parameters, for example, flow and capacity, typical problems are optimising other dependent parameters associated with the network, e.g. total cost, maximal flow, minimal circulation, etc.. In this context, topological and graph-theoretic properties of the network can play a more prominent role.

(N) In Neural networks, nodes represent neurons with the capacity to fire on to their targets with adjustable weights on the edges. The main emphasis is on training the network, i.e. determining appropriate weights, through a variety of inputs for tasks in a given application (see D'Avila Garcez, Lamb, and Gabbay 2008).

(E) In an Ecological setting, the network represents an ecology, where the nodes represent species and the arrows represent dependencies between species. An attack arrow signifies the source species being detrimental to the target species, e.g. source as predator, and the support arrow represents the source being beneficial to its targets, e.g. food. The parameters on nodes can represent relativised population numbers. The parameters on the edges denote interaction parameters between the species relevant to appropriate governing dynamical equations, e.g. the Logistic equation (May 1976) or the Lotka–Volterra equation for predator–prey modelling (Lotka 1925; Volterra 1926). One of the major problems in this area is to identify solutions, e.g. steady state, oscillatory, chaotic, etc., dependent on initial conditions. Clearly, in this model, there is already temporal dependence, even when the parameters do not change over time.

This paper generalises argumentation networks in several directions.

(1) It allows for nodes in argumentation networks not only to attack other nodes but also for support of other nodes. Moreover, we allow for varying strengths of attack and support. We further generalise the model by allowing for strengths of attacks or support themselves to be subject to attack or support.
(2) It allows for the strengths of attack or support to be time-dependent.
This enables us to model the phenomenon of 'Let's lie low and wait for the argument to blow away'.
(3) This paper also examines loop-resolution in argumentation networks, and explores similarities between such loops and predator–prey models in mathematical biology/ecology. One important outcome of such similarity is the possibility of getting extensions by choosing elements in a loop as starting points, assuming they are in, and propagating their attack recursively. In traditional Dung argumentation, this is how we can get the grounded extension. We start with the elements that are not attacked at all, these have to be in, and then we propagate their attacks recursively (see Section 2).

The plan of the paper is as follows:

Section 2 will discuss 'attack only' networks. There are three problems to be addressed in such networks. *Note that we study these problems in the numerical context only.*

(1) The formal definition and motivation of a variety of attack networks.
(2) The modes of attack, a discussion of various options as to how to evaluate the result of attacks.
(3) The resolution of attack loops, such as Figures 4 and 5.

Section 3 is devoted to various methods for the resolution of attack loops. In the course of deciding how to handle loops, we explore formal connections between evaluating local loops occurring in numerical argumentation networks and determining steady-state solutions to network models occurring in mathematical biology/ecology.

Section 4 deals with numerical networks that allow for both attack and support arrows. We quickly ascertain the need to redefine the way in which attack and support are (numerically) carried out, and our considerations lead us to a surprising connection with the Dempster–Shafer rule and with the cross-ratio and projective metrics in geometry.

Section 5 deals with the time-dependent attack and support of arguments. Here a connection with artificial intelligence time–action models is established, as well as a connection with dynamical systems and general temporal logics. If our arguments get numerical values, then in a temporal setting, these values become time-dependent and therefore their rate of change is relevant to the strength of attack.

Section 6 discusses the results and indicates future directions of this work.

2. Attack-only networks with strength

In the 2005 version of this paper, (Barringer et al. 2005), this section was innovative in introducing the systematic study of numerical attacks. As we said before, the argumentation community overlooked the paper (Barringer et al. 2005) for some time and several authors introduced numerical features independently. We shall offer a comparison and discussion of these papers in the conclusion section. However, there are still features in this section which are new and apply equally to numerical and traditional Dung networks. We shall highlight and expand the discussion of these features. They have to do with the dynamic way in which we calculate extensions.

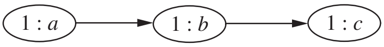


Figure 6. Figure 3 regarded as a numerical network.

To explain this in principle, let our starting point be Figure 3. Viewed as a traditional Dung argumentation network, we can calculate the grounded extension as follows:

Inductive algorithm

Step 1. a is in because it is not attacked.

Step 2. Let all x that are in execute their attack and take out their targets. In our case we get that b is out.

Step 3. Mark as in all nodes that are now not attacked. In this case c is in.

Step 4. Repeat step 2 for the new nodes that are in.

We continue until there is no change. In our case, we get $a =$ in, $b =$ out, $c =$ in. In the numerical form, we get $a = 1$, $b = 0$, $c = 1$.

If we regard this network as numerical in our sense, we need to give the nodes initial strength. Let us choose as an example $V(a) = 1$, $V(b) = 1$, $V(c) = 1$. We get Figure 6.

To remain in the Dung argumentation world, we ignore the numbers suggested by V.

The innovative idea of this section is to use them.

Numerical inductive algorithm

Step 1. a and b and c are in because $V(a) = V(b) = V(c) = 1$.

Step 2. Let all x that are in execute their attack and take out their targets. We get b is out and c is out.

So the final extension we get is $a =$ in, $b =$ out, $c =$ out or numerically we get V' with $V'(a) = 1$, $V'(b) = V'(c) = 0$.

This is what we do in Example 2.3 for the case of a much more complex network.

We now continue this section with an example motivating and explaining the idea of the strength of a node and the strength of attack on a node.

Example 2.1 Consider the election for Governor of California and the then candidate, actor Arnold Schwarzenegger. Let

$a =$ The candidate is alleged to have a certain attitude towards women, and to have behaved towards them accordingly.
$b =$ The candidate will run California very well.

These arguments may have different strengths based on evidence for case a and training and experience for case b. There is also another argument concerning the question of to what extent can argument a attack argument b. Is a relevant at all to b and to what degree? We represent this situation by the network in Figure 7.

The value $\varepsilon = \varepsilon(ab)$, where (ab) is the attacking arc from a to b, represents the strength of the argument that a is relevant to b. It, therefore, can also be attacked, since one can argue against any connection between a and b.[3]

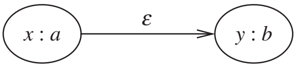


Figure 7. Attacks with different strengths.

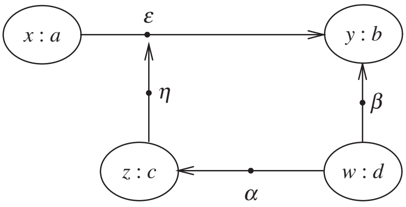


Figure 8. A more elaborate example of numerical network.

The perceptive reader might wonder how the two categories of weights (one on the nodes, one to the links) can be meaningfully instantiated for argumentation purposes.

The idea of associating some sort of strength with an argument (a weight on the nodes), has been proposed by some other approaches to argumentation, see for example Dunne et al. (2009, 2011).

However, we may still ask, what do the weights on the links represent? If it is intended as some sort of contextual impact, then following Martin Caminada views, this factor is in fact grounded in interrelations between the content of arguments, which in reality are structured and sometimes quite involved assertions, abstracted into atomic terms for argumentation frameworks purposes, see for example our paper Gabbay and d'Avila Garcez (2009).

Can the weights attached to the links adequately capture this?

Our answer is no, numbers cannot adequately capture contextual relevance of attacks, (but see Hajek et al. 1992). However, this paper is about numerical networks and under this approach, this is the best that can be done.

Consider the situation described in Figure 8 where argument a has strength x. It attacks argument b, which initially has strength y.

ε is the transmission factor, weakening b in a way that takes account of $x : a$.

b is also attacked by d with factor β.

However, factor ε is attacked by argument c, which is itself attacked by d, with transmission factor α.

This model has two innovations.

(1) The strength of nodes and the transmission factor.
(2) The idea that the transmission factor can itself be attacked. This is the so-called higher level attack,[4] see Gabbay (2009a) for discussion and further references. A restricted form of this idea was later independently discovered by Modgil (2009) and put to good use.

What kind of network does Figure 8 represent? First, note that the strength of nodes is actually a colouring of them. One might expect us to introduce a transmission factor between colours, then in Figure 7 ε could depend only on x and y. We choose to make ε depend on the nodes, taking into consideration that the transmission factor depends on the nature of the argument and not just on their strengths.

The option of attacking transmission factors enables us to delete attacks, one by one, by attacking (lowering) their transmission factor.

Example 2.2 (Modes of attack) Consider a simple numerical model. Assume all values are between 0 and 1. If a is an argument of strength x which is attacking an argument b of strength y, and the transmission rate is ε, then we get εx as the value transmitted.[5] The question now is how does this value εx reduce the value y of b to a new value y'? We have two options. The first is that

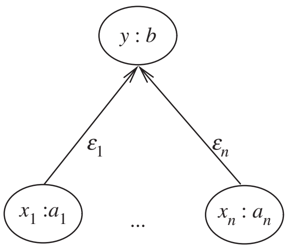


Figure 9. General pattern of attack in numerical network.

the attack reduces the value y of b in proportion, i.e. by εx. Thus, the new value of b is $y(1 - \varepsilon x)$. The second option is that the new value of y is $y' = \varepsilon xy$. This second option makes sense if we view the attack of a on b as a pre-emptive protective measure, reducing a possible attack of b on a. If a is strong ($x = 1$) and $\varepsilon = 1$ then $1 - x\varepsilon = 0$ whereupon a destroys b. This is the previous option, being a genuine attack. However , $\varepsilon xy = y$ when $\varepsilon = 1$ and $x = 1$; so b is not affected. But if x is small, then $y' = \varepsilon xy$ is small. So if b attacks a with transmission rate η, the value of this attack would be $1 - \eta y'$ and the attack would not be effective. Hence the second option can be used as a pre-emptive attack.

We now address the problem of combining attacks. In Figure 8, b is also attacked by d and this attack alone will reduce the value of b to $y(1 - \beta w)$. How do we combine them?

Here too there are two options:

(1) Perform the operation of reduction consecutively (and commutatively), so that the new value of b after the joint attack is $y(1 - \beta w)(1 - \varepsilon x)$.
(2) Add the two reductions, in which case, the new value for b is the value $\max\{0, y - y\varepsilon x - y\beta w\} = \max\{0, y(1 - \varepsilon x - \beta w)\}$.

The advantages of option 1 are that it is simple and that the combination is independent of how the attack is calculated. Another major advantage is that it is compatible with ordinary Dung argumentation when the numerical values are restricted to $\{0, \frac{1}{2}, 1\}$. See Remark 1.2. For example, this can give as the new value of b the combination $\varepsilon x(1 - \beta w)$.

Example 2.2 above has put forward just one mode of attack. There are many other possible modes. Additional possibilities will be examined in Section 4, in conjunction of models with both attack and support. For non-numerical logical modes of attack, see Gabbay and d'Avila Garcez (2009) and Caminada and Gabbay (2009).

In general, we have the situation shown in Figure 9. In this case, we require the following function: If b has value y and if $x_1 : a_1, \ldots, x_n : a_n$ attack $y : b$ with strengths $\varepsilon_1, \ldots, \varepsilon_n$ resp., then we need a function $\mathbf{f}$ such that the new value of node b is $y' = \mathbf{f}(y, x_i, \varepsilon_i)$. This situation is reminiscent of Bayesian networks, where $\mathbf{f}$ is the conditional probability of b on $a_1, \ldots, a_n$.[6]

We adopt option 1 as our mode of attack mainly because of its compatibility with ordinary Dung networks as discussed in Remark 1.2. So the new value $y' = V(b)$ in Figure 9 is

$$\begin{aligned} y' &= y(1 - \varepsilon_1 x_1) \cdots (1 - \varepsilon_n x_n) \\ &= y \prod_i (1 - \varepsilon_i x_i). \end{aligned}$$

The magnitude $\Delta^- y$ which y decreases is

$$\Delta^- y = y - y' = y \left(1 - \prod (1 - \varepsilon_i x_i)\right).$$

Example 2.3 We calculate the transmission of values in Figure 8.

This is to be done in steps by calculating a valuation function V on the nodes of Figure 8. At step $n, n \geq 1$, we define a partial valuation function V_n on the nodes, with some nodes being declared as having a final updated value. Our initial starting function is V_0, with $V_0(a) = x$, $V_0(b) = y$, $V_0(c) = z$, $V_0(d) = w$, $V_0(ab) = \varepsilon$, $V_0(db) = \beta$, $V_0(dc) = \alpha$ and $V_0(c(ab)) = \eta$. V_0 records the values given by the network graphs in Figure 8.

Step 1 : The final updated value V_1 of node d is w, as it is not attacked by anything. Write $V_1(d) = w$. Similarly $V_1(a) = x$. We write V_1 because this is the final value obtained in Step 1.

Step 2 : The new value V_2 of nodes c and b are $V_2(c) = z(1 - \alpha w), V_2(b) = y(1 - \beta w)$. Of course since nodes a and d have already obtained their final value, we can write: $V_2(a) = V_1(a), V_2(d) = V_1(d)$. Node a cannot transmit, at this time (step 2), because we know from the figure that ε is being attacked, and so we need to wait for its value to change. Only when $V(ab)$ gets its final value will a be able to transmit. Thus node (ab) cannot get a final value at this step.

Step 3 : The new value V_3 of the transmission connection (ab) is

$$\begin{aligned} V_3(ab) &= \varepsilon(1 - \eta V_2(c)) \\ &= \varepsilon(1 - \eta z(1 - \alpha w)). \end{aligned}$$

Of course, $V_3(a) = V_2(a), V_3(d) = V_2(d)$, $V_3(c) = V_2(c)$, and $V_3(b) = V_2(b)$.

Step 4 : Now node a can transmit to node b. This gives

$$\begin{aligned} V_4(b) &= V_2(b)(1 - V_3(ab) \cdot x) \\ &= y(1 - \beta w)(1 - \varepsilon x(1 - \eta z(1 - \alpha w))). \end{aligned}$$

Of course, $V_4(a) = V_3(a), V_4(d) = V_3(d), V_4(c) = V_3(c)$ and $V_4(ab) = V_3(ab)$.

Note that node b has had its value changed in bits and pieces. First, it was changed in Step 1 and then in Step 4. This is all right for the current way of changing values, because it is commutative and cumulative. However, the general definition will not allow for this!

This kind of model contains the traditional one as a special case, where all values are taken to be 1 and where there are no attacks on transmissions. Let us see what Figure 8 becomes in this case. Consider Figure 10 and note that it reduces to Figure 11.

We can now give a definition of value propagation for acyclic networks. The general treatment of cycles, or loops, in networks will be addressed in Section 3.

To give a definition, we need to agree on the representation of the network. Let us do it for the case of Figure 8. We need a set of atomic nodes A. In the case of Figure 8, $A = \{a, b, c, d\}$.

To represent the attack of atomic x on y, i.e. the arrow from x to y, we write the expression $x \twoheadrightarrow y$ (called *attacks*).[7] In Figure 8, we have attacks $a \twoheadrightarrow b, d \twoheadrightarrow c$ and $d \twoheadrightarrow b$.

These arrows represent the attacks from a to b, d to c and d to b, respectively. One of these attacks, namely $a \twoheadrightarrow b$, is itself attacked by c. This is represented by the expression $c \twoheadrightarrow (a \twoheadrightarrow b)$.

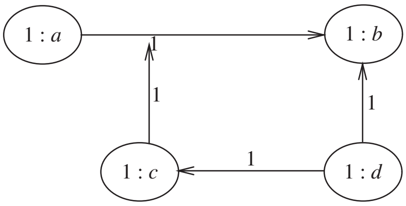


Figure 10. Figure 8 with strength 1.

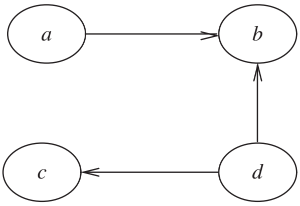


Figure 11. Simpler form of Figure 10.

Note that we *cannot* write an expression of the form $(x \twoheadrightarrow y) \twoheadrightarrow z$. This would mean that the fact that there is an attack from x to y is in itself an attack on z.[8] We are not saying that such reasoning does not exist. We deal with it in the context of fibring networks, see Gabbay (2009c), Gabbay, D.M (1996), Gabbay, D.M. (2009b). In other words, a whole network can be embedded as a node and attacks another node.

Figure 8 can be represented by a set of nodes and attack arrows.

$$T = \{a, b, c, d, a \twoheadrightarrow b, d \twoheadrightarrow b, d \twoheadrightarrow c, c \twoheadrightarrow (a \twoheadrightarrow b)\}.$$

Note that this set T has the property that if $x \twoheadrightarrow y \in T$, then $x \in T$ and $y \in T$. What we still need are the numbers (valuations) in the figure. This we can be represented by a function $V : T \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers.

We are now ready for a formal definition.

Definition 2.4 [9] Let A be a set of atomic nodes.

(1) Define the notion of an attack arrow based on A as follows:
- $a \twoheadrightarrow b$ is an attack arrow if $a, b \in A$.
- $a \twoheadrightarrow x$ is an attack arrow if $a \in A$ and x is an attack arrow.

(2) Let T be a set of attack arrows and atomic nodes. We say that T is an attack network if the following holds
- $x \twoheadrightarrow y \in T$ implies $x \in T$ and $y \in T$.

We say that T is finitely branching (in the outgoing direction) if for every $t \in T$ $\{a | (a \twoheadrightarrow t) \in T\}$ is finite.

(3) A valuation function on T is a function $V : T \to \mathbb{R}$.

(4) An attack network with a valuation is a triple $N = (A, T, V)$, where A is a set of atomic nodes, T is an attack network based on A and V is a valuation on T.

(5) Let $\mathbf{f}$ be a functional giving for each string of real numbers of the form $(y, x_1, \dots, x_n, \varepsilon_1, \dots, \varepsilon_n)$ a new real number $y' = \mathbf{f}(y, \bar{x}_i, \bar{\varepsilon}_i)$ (where $\bar{z}_i$ abbreviates $z_1, \dots, z_n$, for $z = x$ or $z = \varepsilon$). Note that n is arbitrary. We assume $\mathbf{f}$ to be continuous and symmetrical in the pairs of variables $(x_i, \varepsilon_i), i = 1, \dots, n$ and generally nice.[10,11]

For example, let $\mathbf{f}(y, \bar{x}_i, \bar{\varepsilon}_i) = y \prod_{i=1}^{n}(1 - \varepsilon_i x_i)$. See Section 4.2 for more options.

(6) An argumentation attack model is a pair $(N, \mathbf{f})$, where N and $\mathbf{f}$ are as above.

Example 2.5 Let us look at some examples. Consider Figure 12 in which a attacks b but also attacks its own attack. This is a case of a self defeating attack of a on b.

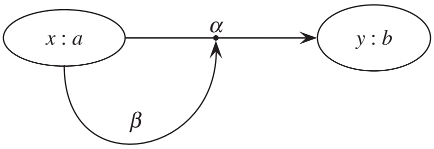


Figure 12. A form of self attack.

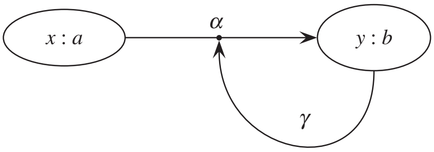


Figure 13. A form of self defence.

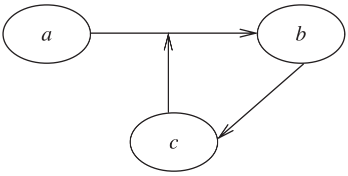


Figure 14. Cycling self attack.

We have $T = \{a, b, a \twoheadrightarrow b, a \twoheadrightarrow (a \twoheadrightarrow b)\}$ and

$$\begin{aligned} &V(a) = x, V(b) = y, \\ &V(a \twoheadrightarrow b) = \alpha \quad \text{and} \\ &V(a \twoheadrightarrow (a \twoheadrightarrow b)) = \beta. \end{aligned}$$

We can compare Figure 12 with Figure 13. In Figure 13, we can interpret γ as a feedback loop, attacking and reducing α. The weaker the argument b is, the less we want to spend the effort to attack it.

Definition 2.6 (Syntactic acyclicity[12]) Let T be an attack network. Define $R_T \subseteq A^2$ as follows:

$$aR_Tb \text{ iff } a \twoheadrightarrow b \in T \text{ or for some } x \in A, a \twoheadrightarrow (x \twoheadrightarrow b) \in T.$$

Let R_T^* be the transitive closure of T. We say T is syntactically acyclic iff there is no $x \in A$ such that xR_T^*x.

If $N = (A, T, V)$, we say N is syntactically acyclic if T is such.

Example 2.7 Figure 14 is cyclic while Figure 15 is acyclic and finitely branching.

Example 2.8 Figure 16 is cyclic syntactically, but is acyclic after evaluation using V.

Note that although the network is syntactically cyclic, since $V(\beta) = 0$, it is as if $b \twoheadrightarrow a$ does not exist in T. We shall deal with this kind of *semantic* acyclicity later.

Definition 2.9 (Value propagation) Let $(N, \mathbf{f})$ be a model, where N is syntactically acyclic and finitely branching. We shall propagate the values V through the model using $\mathbf{f}$. We do this in waves.

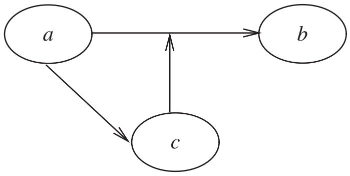


Figure 15. Acycling self attack.

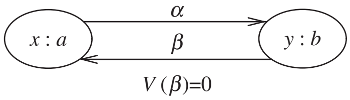


Figure 16. Syntactically cycling figure, but not semantically cycling.

Wave m will define values $V_m(x)$ for some $x \in T$ and x is then referred to as an updated element with the updated value $V_m(x)$.

Wave 0

An element $a \in T$ is said to be syntactically free of attack if for every $e \in A$, we have $(e \looparrowright a) \notin T$. Let it be said that the updated elements of Wave 0 are free of attack elements and let the updated value V_0 be $V_0(a) = V(a)$, for an updated a of wave 0.

Wave $n + 1$

Assume we have defined the updated elements of waves $k \leq n$ and their updated value V_k. Let b be any element and let $a_1, \dots, a_m$ be all, if any, elements of T such that $(a_i \looparrowright b) \in T$. Assume for each i, that a_i, as well as $a_i \looparrowright b$, were updated at some earlier wave $k_i \leq n$ and $l_i \leq n$, respectively.

Define

$$V_{n+1}(b) = \mathbf{f}(V(b), \bar{V}_{k_i}(a_i), \bar{V}_{l_i}(a_i \looparrowright b)).$$

When the network is finite, the algorithm updates all the nodes and terminates with some $V_n = V'$ in quadratic time.[13]

Example 2.10 (Figure 8 revisited) Let us examine the network of Figure 8 again. We are listing the updated elements. Let us compare with Example 2.2.

Wave 0

$$w : d, x : a, \beta : d \looparrowright b,$$
$$\alpha : d \looparrowright c, \eta : c \looparrowright (a \looparrowright b).$$

Wave 1

$$z(1 - \alpha w) : c.$$

Note that the only updated element in this wave is c. b is not updated because not all of its attackers (namely a) have been updated. In our earlier computation, we did attack b at this stage, but we cannot do that under our current definition. We will not get a different result because our function $\mathbf{f}$ launches the attacks from separate nodes independently, cumulatively and commutatively.

Wave 2

$$\varepsilon(1 - \eta z(1 - \alpha w)) : a \looparrowright b.$$

Here $a \looparrowright b$ is being updated.

Wave 3

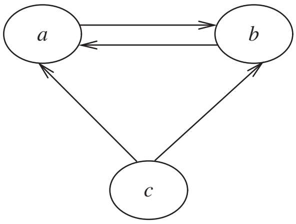


Figure 17. Semantically cycling figure but not syntactically cycling.

Now we can update b. We get

$$y(1-\beta w)(1-\varepsilon x(1-\eta z(1-\alpha w))) : b.$$

Definition 2.11 Let $(N, \mathbf{f})$ be a finite model. Propagate V using $\mathbf{f}$ in waves as defined above. Let the new valuation $V'(a), a \in T$, be the updated value of a. We call V' the result of the waves of attack in the network. Note that the propagation is executed only once.

Remark 2.12 (Networks with values in $\{0, 1\}$) For such networks more can be said. Consider the situation in Figure 17

Although the network is not syntactically acyclic, it is what we can call semantically acyclic. Let us propagate the annotations in recursive steps which we shall call waves.

Wave 0

The element c is updated to having value $+1$, since it is not attacked by anything.

Wave 1

The nodes a and b are attacked by c and therefore updated to value 0. We have a clear updated semantic solution V' in this case.

This works only when the functions $\mathbf{f}$ can give value to arguments which are not evaluated themselves. Let us do this formally.

We have $V_0(c) = 1$ and just let $V_0(a) = x, V_0(b) = y$ where x and y are any values.

Wave 0

$V_0(c) = 1$ because c is not attacked.

Wave 1

$$\begin{aligned} V_1(c) &= V_0(c), \\ V_1(a) &= \mathbf{f}(x, y, 1) = 0, \\ V_1(b) &= \mathbf{f}(y, x, 1) = 0. \end{aligned}$$

We accept $V_1(a), V_1(b)$ as final updated values because for the value $V_0(c) = 1$, the function $\lambda x \lambda y \mathbf{f}(x, y, 1)$ is identically 0, no matter what x, y are.

Definition 2.13 (Semantic acyclicity Version 1) Let $(N, \mathbf{f})$ be a model where N is finitely branching but not necessarily syntactically acyclic. We say that $(N, \mathbf{f})$ is semantically acyclic Version 1 if we can propagate V in the waves as follows:

Wave 0 An element b is said to be semantically (Version 1) 'free' of attacks if either (a) or (b) hold.

(a) b is syntactically free of attack, i.e. for every $e \in A$ we have $e \looparrowright b \notin T$. In this case we set $V_0(b) = V(b)$ as the updated final value of b.

(b) Let $a_1, \ldots, a_m$ be all elements of A such that $a_i \looparrowright b \in T$. Assume that $y = \mathbf{f}(V(b), V(a_i), V(a_i \looparrowright b))$ depends on $V(b)$ only. Then let $V_0(b) = y$ be the updated value of b.

Wave $n + 1$

Assume that we have defined the updated elements of waves k for all $k \leq n$. Let b be any element such that for some $a_1, \ldots, a_m, a_i \looparrowright b \in T$ for $i = 1, \ldots, m$. Consider the value $y = \mathbf{f}(V(b), x_1, \ldots, x_m, w_1, \ldots, w_m)$.

Assume that for any i, j such that the value x_i of a_i or w_j of $a_j \looparrowright b$, respectively, has *not* been updated at some earlier stages k_i or $l_j \leq n$, we have that y does *not* depend on x_i, w_j, respectively.

Then we say that b is updated to the value $V_{n+1}(b)$ at stage $n + 1$ and the value is y, where:

$$y = \mathbf{f}(V(b), V_{k_i}(a_i), V_{l_j}(a_j \looparrowright b)),$$

where $V_{k_i}(a_i), V_{l_j}(a_j \looparrowright b)$ are the updated values of a_i and respectively $a_j \looparrowright b$ if such values exist and otherwise we do not care. We can assign this value to y because we assumed that y does not depend on the nodes that were not updated.

If the algorithm terminates giving updated values to all nodes of T, we say the network is semantically acyclic Version 1.

Example 2.14 Let us check some of our previous figures for semantic acyclicity. Figure 4 is cyclic because, for example,

$$\mathbf{f}(V(b), V(a)) = V(b)(1 - V(a))$$

depends on $V(a)$ unless $V(b) = 0$. Similarly, Figure 5.

Example 2.15 Consider the situation of Figure 12. Assume that in this figure all initial values are 1, i.e. $x = y = \alpha = \gamma = 1$. So $V \equiv 1$.

Let us try to update by waves and see what happens.

Wave 0

The final updated value for a in this wave is $V_0(a) = 1$.

Wave 1

We cannot get a value for b, even semantically Version 1, because the value of b is

$$V_1(b) = V(b)(1 - V_0(a)V_0(a \looparrowright b)).$$

This value depends on the value of $V_0(a \looparrowright b)$ and this value is not finally updated.

So we cannot continue. However, there is something we can do. The initial value $V(a \looparrowright b)$ is 1. So we can propagate the attack from a to b and get a new value $V_1(b) = 0$. With this value, b cannot attack $a \looparrowright b$ and so the initial value 1 of $a \looparrowright b$ remains unchanged. So we do get a stable situation and not an oscillating situation.

This example suggests another definition of semantic acyclicity, a new Version 2.

Definition 2.16 (Semantic acyclicity Version 2) Let $(N, \mathbf{f})$ be a model where N is finitely branching but not necessarily syntactically acyclic. We say that $(N, \mathbf{f})$ is semantically acyclic Version 2 if we can propagate V in waves as follows:

Wave 0

Let b be an element which is semantically free of attack according to Version 1 as in wave 0 of Definition 2.13. Let b be updated to $V_0(b) = V(b)$.

For any other element c which is not free let $V_0(c)$ be undefined.

Wave $n+1$

Assume V_n has been defined, giving values to some elements of T. Further assume that some of these values are declared final and the rest of these given values are declared temporary. For some elements of T V_n may not give a value at all.

Let b be any element such that for some $a_1, \dots, a_m, a_i \looparrowright b \in T, i = 1, \dots, m$.

Consider $w = V_m(b), x_i = V_n(a_i)$ and $y_i = V_n(a_i \looparrowright b), i = 1, \dots, m$.

Some of these values are final, some are temporary and some are undefined because V_n does not give a value. We assume that b is such that for some a_i or some $a_i \looparrowright b$, V_n does give some value. If V_n does not give any value to any of a_i and to any of $a_i \looparrowright b$, we do not touch b at this wave. This condition we call the slow propagation condition and it is intended to avoid un-intuitive situations like the one in Example 2.17.

So we can assume that for some of w, x_i or $y_i, i = 1, \dots, m, j = 1, \dots, m$, V_n does give a value final or temporary. To be in a situation where all variables w, x_i, y have defined values, let the value for w, x_i and y_j be $V(b)$, $V(a_i)$ or $V(a_j \looparrowright b)$ in case V_n is not defined and does not give them values.

Thus we can now calculate

$$z = \mathbf{f}(w, x_1, \dots, x_m, y_1, \dots, y_m).$$

We distinguish three cases.

Case 1

V_n does not give a value for b : In this case, declare $V_{n+1}(b) = z$ and declare this value as temporary.

Case 2

$V_n(b) = w$ and $z = w$: In this case, reassert $V_{n+1}(b) = V_n(b)$ as final or temporary, giving b the same status as that which we have for V_n.

Case 3

$V_n(b) = w$ and $z \neq w$. In this case, we stop and say that $(N, \mathbf{f})$ is *not* semantically acyclic Version 2.

We say that $(N, \mathbf{f})$ is semantically acyclic Version 2 if for some n, V_n is total and $V_n = V_{n+1}$.

Example 2.17 Consider again the situation in Figure 5. Let us check for semantic acyclicity Version 2 without the slow propagation condition. Assume V gives values 1 and all transmission values are 1.

Wave 0

V_0 cannot be defined on any node.

Wave 1

We use the values of V, since V_0 gives no values and since b is attacked by a we get that its V_1 temporary value is 0. Similarly, $V(a) = 0, V_1(c) = V_1(d) = 0$. All temporary values.

Wave 2

The values remain 0 and so the network stabilises at $V_2 \equiv 0$.

The above calculation was without slow propagation.

If we invoke the slow propagation condition, Wave 1 cannot be executed. So we get no values.

Example 2.18 Consider the situation of Figure 14. Let us test for semantic acyclicity Version 2. Assume all transmission values and all node values are 1, i.e. $\lambda x V(x) \equiv 1$.

Wave 0

We have $V_0(a) = V(a) = 2$. We cannot have any more values.

Wave 1

We have $V_1(b) = 0$ because $V_0(a) = 1$ and we use the transmission value $V(a \looparrowright b) = 1$. We cannot get a temporary value for c because we do not have any value for b at this wave.

Remark 2.19 Let us assess what we have so far. Consider a network $(N, \mathbf{f})$ and consider maximal cycles with respect to the relation R_T of Definition 2.6. If $\mathbb{C}$ is such a cycle, then if there is an element out of the cycle with an arrow into it, then there is a chance for the waves of the evaluation of Definition 2.16 to semantically resolve the cycle. The problem arises when the cycle $\mathbb{C}$ has no attack arrows leading into any of its members. Definition 2.16 will not be able to do anything, as we see in Example 2.17, which deals with the situation in Figure 5. We, therefore, need a new idea or methodology for resolving complete cycles and giving some values to its nodes. This new idea and these new values come from the treatment of loops in ecological systems and we address this in Section 3.

Remark 2.20 (Equational labelling) Consider the special case of a network with no higher level attacks. This is just a set S with a binary relation on it. We allow for nodes to have strength as real values between 0 and 1 inclusive, and allow for full transmission. In terms of the basic situation described in Figure 9, we have $\varepsilon_1, \dots, \varepsilon_n$ are all 1 and the value of y is given by the formula

$$y = e(x_1) \cdots e(x_n), \tag{1}$$

where the function e is defined as $e(x) = 1 - x$.

Note that $y = 1$ only if all the x_i are 0.

We can ask for a function V giving values to nodes and satisfying the equation above for any instance of Figure 9 in the network such that $a_1, \dots, a_n$ are ALL the nodes attacking b. We can use linear programming methods to find such a V.

This approach was introduced in Gabbay (2009c) and fully developed in Gabbay (2011a, c), and generalises the Caminada labelling approach (Caminada 2011). It has two advantages

(1) We can let V take values in any commutative and associative multiplicative algebra. So, for example, the {0, 1, ?} valued Caminada labelling can be viewed as equational labelling in the three valued commutative associative algebra satisfying the axioms

$$x1 = 1x = x$$
$$x? = ?x = ?$$
$$0x = x0 = 0.$$

We let $e(1) = 0$, $e(0) = 1$ and $e(?) = ?$ (see Caminada and Gabbay 2009; Gabbay 2009c).

(2) We can write general non-local equations or constraints which V should satisfy and seek a solution. This allows for more general networks. Equation (1) is local, it involves a point and its immediate neighbours, it is not the most general type of equation.

Note that Figure 4 has the $a = b = c = ?$ solution in the three-valued algebra and the $a = b = c = \frac{1}{2}$ in the real numbers.

3. Handling loops – ecologies of arguments

This section is about handling loops, both in numerical argumentation networks and, in general, numerical networks, especially ecological networks.

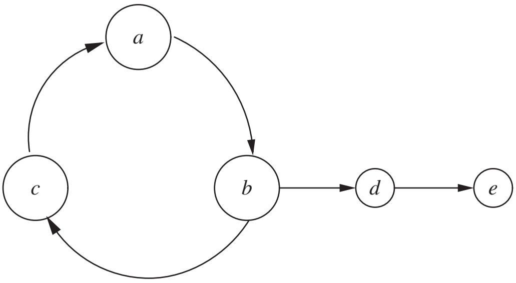


Figure 18. A frustrating initial loop.

3.1. *Loops in argumentation networks*

In the case of argumentation, the situation is quite simple. We make a distinction between even and odd loops. The even loops can give rise to non-trivial extensions, while the odd loops can only give rise to the trivial extension all undecided.

The situation can be frustrating when we have an odd loop at the top of a network (i.e. none of the members of the loops is being attacked from outside the loop). Figure 18 is a typical situation.

The only extension we can have here is all undecided, because the top odd loop $\{a, b, c\}$ is a bottleneck.

Recently, authors looked for ways of breaking such bottlenecks by putting forward new semantics (Baroni and Giacomin 2003; Baroni, Giacomin, and Guida 2005; Bodanza and Tohmé 2009; Gaggl and Woltran 2010; Abraham et al. 2011a,b). The CF2 semantics, for example, will resolve the loop of Figure 18 by taking maximal conflict-free subsets of the loop instead of admissible subsets. This would give us the following CF2 extensions:

$$\{b, e\}$$
$$\{a, d\}$$
$$\{c, d\}.$$

When we take a numerical point of view, we get equations to solve, and the solutions to the equations are the extensions. The equations in this case are (adopting option 1 discussed in Section 2 after Example 2.2) as follows:

(1) $V(a) = 1 - V(c)$
(2) $V(b) = 1 - V(a)$
(3) $V(c) = 1 - V(b)$
(4) $V(d) = 1 - V(b)$
(5) $V(3) = 1 - V(d)$.

The only solution is

$$V(a) = V(b) = V(c) = V(d) = V(e) = \tfrac{1}{2},$$

which corresponds to all undecided. So the problem is not solved, but is less frustrating, as we have more scope of writing modified equations, getting new solutions.

The traditional Dung approach gives a limited number of parameter concepts to play with. We have conflict-free, skeptical, credulous, attack, defence, etc., and can modify these to get ourselves out of odd loops. In the numerical approach, we have the entire landscape of (what is known in

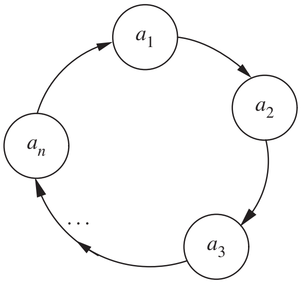


Figure 19. General loops.

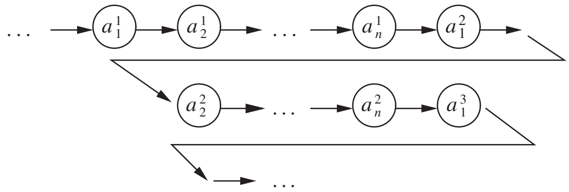


Figure 20. Unfolding a general loop.

fuzzy logic as) De Morgan norms to use. See Gabbay (2011c) for a wider discussion. The De Morgan norms keep compatibility with the traditional Dung approach to argumentation networks.

This numerical point of view certainly connects with other types of networks, such as ecological networks. The equations are different there, not compatible with argumentation, but the scope for resolving loops is much richer, as we discuss in this section.

We add one more comment about unfolding loops. Consider the loop in Figure 19. In this figure n may be odd or even. When we unfold it, we get Figure 20.

There is no distinction in Figure 20 between odd or even, except that by writing copies a_k^i of the letter a_k, we indicate that we want either all a_k^i for all i to be in or all a_k^i for all i to be out.

With this restriction, if n is even, there is a solution and if n is odd, there is no solution.

Now that we have a general orientation between loops in traditional Dung argumentation and loops in general numerical networks, we can start the detailed discussion of this Section. We have encountered loops in Example 1.1(b) and (c). In Figure 5 of (b), we need to resolve the loop $\{?a, ?b\}$ in order to propagate values to c. So technically all we need is some loop-resolving compromise assignment of values to a and to b, and then the algorithm of Definition 2.9 can be invoked. How do we obtain such a value?

The values we give to the loop depend on our interpretation of it. Hints for possible interpretations can be obtained from other possible interpretations of the entire network regarded as a mathematical entity. We shall, therefore, open this section by putting forward several points of view as to the meaning of labelled networks and their internal loops, which will then lead to ways of dealing with their loops.

To begin our discussion, consider the following Figure 21 which is the typical general numerical case of an even loop.

Let $\mathbf{f}_1(y, x, \varepsilon), \mathbf{f}_2(x, y, \eta)$ be the two transmission functions. We observe the following:

(1) Figure 21 describes a syntactical loop.

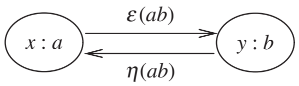


Figure 21. General even numerical loop.

(2) Depending on the values x, y, ε, η and depending on the functions $\mathbf{f}_1$ and $\mathbf{f}_2$, Figure 21 might not be a loop semantically. For example, if $x : a$ is much stronger than $y : b$ or if $\varepsilon = 0$ then this might not be a loop.

The general method is to find a solution for the pair of equations

$$y = \mathbf{f}_1(y, x, \varepsilon)$$
$$x = \mathbf{f}_2(x, y, \eta)$$

or, more generally, for the case of a maximally strongly connected component of the network involving n nodes $x_1, \ldots, x_n$, we solve the vector equation

$$\mathbf{X} = \mathbf{F}(\mathbf{X}, \overline{\Pi}).$$

We can then use a solution (if it exists) as the values of the nodes in the context of the original network. Thus, for the loop of Figure 4 mentioned above, values for the nodes a and b can be determined using this method. We get the value for Figure 4 as $\frac{1}{2}$ for all nodes.

The reader should note that Brouwer fixed the point theorem (see Wikipedia for reference) that guarantees at least one solution to the above equation. There may be many solutions. It is up to us to decide whether we want to generate all solutions or seek a solution with certain properties/constraints (some constraints may not allow for a solution). If we have certain solutions in mind, then special numerical analysis techniques may have to be employed. See Gabbay's (2011a, c) papers. These options in seeking solutions is reminiscent of the discussion in Logic Programming (the so-called great logic programming schism).

Note that in Appendix 2, we turn to possible interpretations of loops in other kinds of networks. We examine these different interpretations in order to assess the relevance of their solutions to the situations arising in argumentation. We will also examine whether the argumentation network point of view may be applied to other networks.

3.2. *Unfolding loops*

There are various ways of treating loops.

(a) We can unfold them as done in, say, modal logic.
(b) We can let node a attack b, calculate the new value and then let b attack a, calculate the new value and then let a attack b and so on. This we call the parasite way of unfolding a loop.
(c) We can let a and b attack each other simultaneously, calculate the new values and then let them attack again and again. This is the predator–prey way of unfolding a loop.
(d) We can assume a steady state and solve the equations suggested by the geometry of the loop.

Let us now turn to Figure 21 and see what are our options for dealing with this loop.

Our first attempt at a solution is to regard (ab) and (ba) as the same channel and read the loop as feedback loops (see Appendix 2). So a pushes εx towards b and b pushes ηy towards a. The

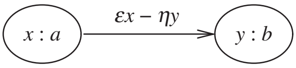


Figure 22. Option 1 for resolving the loop of Figure 21.

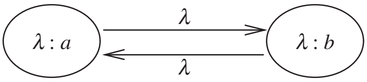


Figure 23. Option 2 for resolving the loop of Figure 21.

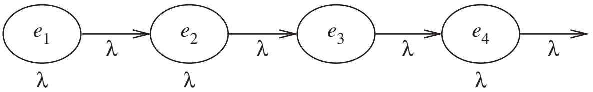


Figure 24. Unfolding Figure 23.

net result is $(\varepsilon x - \eta y)$ in the direction of the positive value. So assuming $\varepsilon x \geq \eta y$, we get that Figure 21 is essentially reduced to Figure 22.

The solution is not satisfactory. It cannot deal with cases like Figure 4 unless we further commit the model to be a proper network flow model with various capacities, as studied in operational research. So let us try another approach. Assume in Figure 21 that we have $x = \eta = \varepsilon = y$.

Call the common value λ. We now get Figure 23.

Let $V_0(a) = V_0(b) = \lambda$, the initial value, and let us transmit from a to b and back from b to a in cycles and see what presents itself. This is the modal logic approach.

We treat Figure 23 as equivalent to Figure 24.

In Figure 24, nodes $e_1, e_3 \ldots$ represent node a of Figure 23 and needs $e_2, e_4, \ldots$ represent node b. So we start from $V_1(e_1) = \lambda$ and transmit to the right getting $V_n(e_n),\ n = 2, 3, \ldots$.

Step 1: Transmit λ^2 to e_2 to get $V_2(e_2) = \lambda(1 - \lambda^2) = \lambda - \lambda^3$.

Step 2: Transmit from e_2 to e_3 the value $\lambda V_2(e_2)$ and get $V_3(e_3) = \lambda(1 - \lambda V_2(e_2)) = \lambda - \lambda^3 + \lambda^5$.

We can continue by induction.

LEMMA 3.1 Suppose we have a node e with $V(e_n) = V_{\lambda,n} = \lambda - \lambda^3 + \lambda^5 - \cdots + (-1)^n \lambda^{2n+1}$ and suppose we are transmitting to a node $\lambda : e_{n+1}$ with value λ then we get $V(e_{n+1}) = V_{\lambda,n+1}$.

Proof

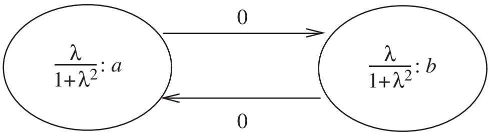
$$\begin{aligned} V(e_{n+1}) &= \lambda(1 - \lambda V(e_n)) \\ &= \lambda - \lambda^3 + \lambda^5 \cdots - \lambda^2(-1)^n \lambda^{2n+1} \\ &= \lambda - \lambda^3 + \lambda^5 \cdots + (-1)^{n+1} \lambda^{2(n+1)+1}. \end{aligned}$$

■

We now observe that when n goes to infinity, we get $V_{\lambda,\infty} = \lambda/(1 + \lambda^2)$.[14]

This means that Figure 23 stabilises into Figure 25. Note that the transmission rates in Figure 25 are all 0. This is because the values $V_{\lambda,\infty}$ obtained have already taken into account all recursive transmissions.

Note that Figure 24 represents one way of going through the cycle of Figure 23, i.e. the fuzzy modal logic approach. Another approach is what we called the parasite model, where we apply



$\frac{\lambda}{1+\lambda^2}: a$
$\frac{\lambda}{1+\lambda^2}: b$
0
0

Figure 25. Stable state of Figure 23.

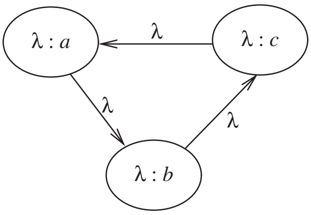


Figure 26. Odd numerical loop.

the transmission on Figure 23 directly, starting from node a to b with $V_1(a) = \lambda$, (corresponding to $V_1(e_2)$) we would get $V_2(b) = \lambda(1 - \lambda^2)$, same as $V_2(e_2)$ and then transmit back to node a and get $V_3(a) = V_1(a)(1 - \lambda V_2(b))$ (corresponding to $V_3(e_2)$). So far, the values agree, but now there is a difference. Working directly on Figure 23, we transmit $1 - \lambda V_3(a)$ to node b whose *last* value is $V_2(b) = \lambda(1 - \lambda^2)$ and get $V_4(b) = \lambda(1 - \lambda^2)(1 - \lambda V_3(a))$. While in Figure 24, the value of node e_4 (which corresponds to b) is λ and so we get in Figure 24, $V_4(e_4) = \lambda(1 - \lambda V_3(e_3))$. So the question is, as we go through the cycle $a \to b \to a \to b \ldots$, do we use the new value or follow Figure 24 and keep the value at λ, the initial value!

Another possibility for dealing with Figure 23 is to adopt the predator–prey model and transmit simultaneously from node a to node b and from node b to node a, and then repeat the cycle. If $V_0(a) = V_0(b) = V_0 = \lambda$ is the initial value, then symmetry is maintained through the cycles and for step $n + 1$, we get

$$V_{n+1}(a) = V_{n+1}(b) = V_{n+1} = V_n(1 - \lambda V_n).$$

So we end up with a recursive equation

- $V_0 = \lambda, 0 \leq \lambda \leq 1$
- $V_{n+1} = V_n(1 - \lambda V_n)$

which for $0 \leq \lambda \leq 1$ gives $V_\infty = 0$, meaning that a and b cancel each other.[15]

The above considerations can be applied to other loops. The net result of Figure 4 will be similar to that of Figure 23.

Consider Figure 26. Let us see what different options for resolving loops yield for the figure.

Option (a), the unfolding option would yield, by using Lemma 3.1 that Figure 26 stabilises as Figure 27

Option (b) yields the sequence of Figure 24 (which is not surprising because of symmetry) and therefore will give rise to Figure 27 again.

Option (c) again because of symmetry yields for each node the sequence of Figure 24. So again we get Figure 27 as the solution of the loop.

Option (d) is perhaps the simplest and most clear of all the options. We assume a steady-state solution for nodes V_a, V_b, V_c. We get the equations

(1) $V_a = V_a(1 - \lambda V_c)$

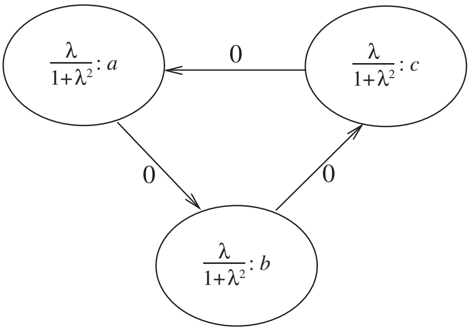

$\frac{\lambda}{1+\lambda^2}: a$
0
$\frac{\lambda}{1+\lambda^2}: c$
0
0
$\frac{\lambda}{1+\lambda^2}: b$

Figure 27. Solution to odd loop.

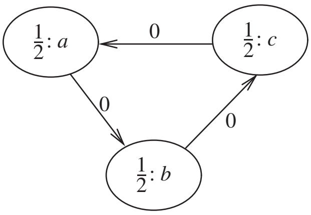


Figure 28. Another solution to odd loop.

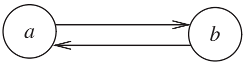


Figure 29. An even loop.

(2) $V_b = V_b(1 - \lambda V_a)$
(3) $V_c = V_c(1 - \lambda V_b)$

Because of symmetry, we can assume

$$V_a = V_b = V_c = V.$$

Thus we get $V = V(1 - \lambda V)$. The only solution is $V = 0$.

We can make one more move now. To resolve Figure 4, we consider Figures 26 and 27 and let λ approach 1. Thus, we get the value $\frac{1}{2}$. Hence the net results of Figure 4 is Figure 28.

A similar net result obtains for Figure 29.

Note that now we can resolve the loop in Figure 5. We get $V(a) = V(b) = \frac{1}{2}$ and therefore $V(c) = \frac{3}{4}$ and hence $V(d) = \frac{1}{4}$.

We can also deal with an argument attacking itself. It will get $\frac{1}{2}$. We pause to remark that for the reader interested in argumentation networks only, the best method for resolving loops is option (d), namely assume a steady-state solution and solve the resulting equation (see Gabbay's 2011a, 2011c). For other types of networks (see Appendix 2) other options may be more natural.

There is still work to be done on resolving loops. We have some relevant papers which handle loops, especially using the equational approach. This is a complex topic. In the context of our paper here, we need to show the following:

(1) How the results we get for the loop depend on the choice of numbers we assign to the nodes and for the transmission rates (we gave λ to all!).

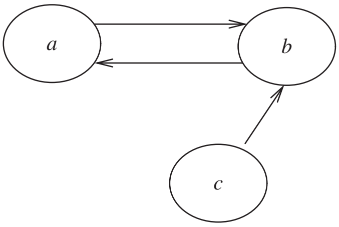


Figure 30. Sample loop.

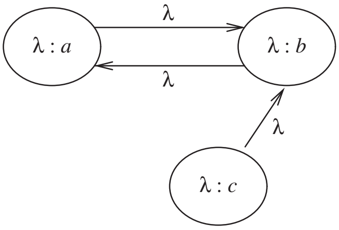


Figure 31. Sample loop.

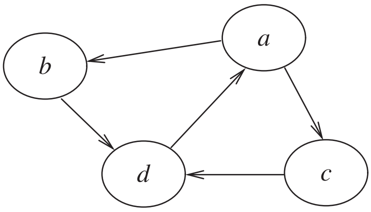


Figure 32. Sample loop.

(2) What happens when loops can be resolved but we use our method anyway, as in Figure 30. In Figure 30, the net result is

$$\{+c, -b, +a\}.$$

What do we get if we assign λ everywhere and get Figure 31?
Here is the calculation: We start with $V_0(a) = V_0(b) = V_0(c) = \lambda$. Transmit from c to b and get $V_1(b) = \lambda - \lambda^3$. Transmit from b to a and get $V_1(a) = \lambda(1 - \lambda V_1(b)) = \lambda - \lambda^3 + \lambda^5$. Obviously, if we follow the loop, we get as before $V_\infty(a) = V_\infty(b) = \lambda/(1 + \lambda^2)$ and the net result is $\{1 : c, \frac{1}{2} : a, \frac{1}{2} : b\}$. This is not satisfactory.
It makes more sense to try to give c value 1 transmitting at rate 1, since c is not in a loop. This will give b value 0 and a value λ. When λ approaches 1 we get the right answer. Perhaps we might follow the procedure of giving λ only to nodes in a loop?

(3) Consider, however, the following loop in Figure 32.
d is attacked twice and is attacking once, while a is attacking twice and is attacked once. Should we give them λ in the same way?

Example 3.2 (Resolving Figure 32) Let us try the fixed point approach on Figure 32. We begin with $V_0(a) = V_0(b) = V_0(c) = V_0(d) = y$ and with transmission λ.

(a) We start propagating from node a. We get

$$V_1(b) = V_1(c) = y(1 - \lambda y).$$

$$V_1(d) = y(1 - \lambda y(1 - \lambda y)^2)$$

and therefore

$$V_1(a) = y(1 - \lambda V_1(d)).$$

We need a fixed point solution to $V_1(a) = V_0(a)$. Hence

$$y(1 - \lambda V_1(d)) = y.$$

Excluding $y = 0$, we get

$$1 - \lambda V_1(d) = 1.$$

Hence

$$V_1(d) = 0.$$

This means

$$y(1 - \lambda y(1 - \lambda y)^2) = 0.$$

Hence

$$\lambda y(1 - \lambda y)^2 = 1.$$

Let $x = \lambda y$. We get $x(1 - x)^2 = 1$. This has a solution, x_0 of approximate value

$$x_0 \approx 1.755.$$

If we want $0 \leq \lambda \leq 1$ then there is no way $0 \leq y \leq 1$. Hence the only fixed point solution is $y = 0$.

(b) Let us start at node d of Figure 32

$$V_0(d) = y$$

$$V_1(a) = y(1 - \lambda y)$$

$$V_1(b) = V_1(c) = y(1 - \lambda V_1(a))$$

$$V_1(d) = y(1 - \lambda V_1(b))^2$$

and try to solve the fixed point equation:

$$y = y(1 - \lambda V_1(b))^2.$$

Hence if we insist on $y \neq 0$,

$$1 = (1 - \lambda V_1(b))^2.$$

Hence

$$1 - \lambda V_2(b) = \pm 1.$$

So either

(i) $V_1(b) = 0$

or

(ii) $V_1(b) = 2\lambda$.

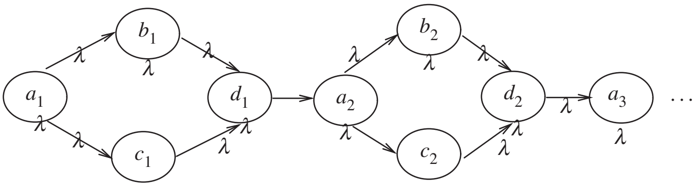


Figure 33. Unfolding Figure 32.

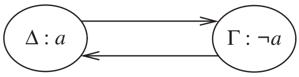


Figure 34. A loop where the annotations are logical theories.

For $V_1(b) = 0$ we get, if $y \neq 0$, that $V_1(a) = 1/\lambda$.

Hence $\lambda y(1 - \lambda y) = 1$. It is clear that this equation has no real solution.

Let us now try the case in which $V_1(b) = 2\lambda$.

Hence

$$y(1 - \lambda y(1 - \lambda y)) = 2\lambda$$

$$y - \lambda y^2(1 - \lambda y) = 2\lambda$$

$$y - \lambda y^2 + \lambda^2 y^3 - 2\lambda = 0.$$

Does this have solutions? Remember $0 \leq \lambda \leq 1, 0 \leq y \leq 1$.

If we choose $\lambda = 0.133$ and $y = 0.275$, the value of the polynomial is 0.0006.

Since we are dealing with continuous functions, we can find proper solutions.

Let us now try another way of tackling Figure 32, which can be rewritten as Figure 33 below, where a_i represent a, b_i represent b, c_i represent c and d_i represent d.

The neural net approach gives us an additional dimension. We can run the cycles in the loop but also transmit to the rest of the network, and possibly stop after so many cycles (say $n = 100$), and examine the values in all nodes of other network. If the time involved in the cycles has meaning in terms of the network itself changing in time (as modelled in Section 4 below), then we have added a new and interesting dimension to loops in these networks.

In other words, we are saying that attacks take time to be executed, a loop of the form '*a* attacks *b* and *b* attacks *a*' also takes time to unfold, and meanwhile the network can change.

To give an example of such a loop, think of contradicting witnesses and circumstantial evidence, one supporting a and one supporting $b = \neg a$. So the loop is as in Figure 34, where Δ, Γ are themselves argument structures which are time-dependent. This loop certainly takes time to unfold! There may be some facts in Δ or Λ that take time to verify or refute!

We conclude this section with a remark which may be of interest to a reader who wants to expand his horizon beyond the area of argumentation networks. We believe that the general treatment of loops should be done in the context of neural networks (see d'Avila Garcez, Gabbay, and Lamb 2004), not because of a conceptual connection, but because these nets can technically reach equilibrium and resolve loops of the kind that arise there.

Note that every graph can be presented as an acyclic graph of nodes which are themselves maximally connected cycles. So when we are dealing with cycles, we can make use of that. In fact Baroni et al. (2005) took advantage of this idea.

4. Attack and support networks

This section deals with attack and support. Since our approach is numerical, all we have is several numbers attacking or supporting another number. From the point of view of traditional argumentation networks, this simplification may appear to be going too far. We lose the richness of discussions and modelling, we have in a substantial number of papers on support which we have recently seen published in the argumentation community. However, outside argumentation, numerical attack and support make sense. In ecological networks, in flow networks, in electrical networks, etc.

So this section is concerned with numerical attack and support. The main problem is what numerical function to use in the case of nodes $x_i : a_i$ support the node $y : b$. What function $y' = \mathbf{g}(y, x_i)$ should give us the new supported value y'? We have this problem in argumentation too, but in our case, it is all numerical.

Because we are dealing with numbers, we add the requirement that if $y : b$ is both supported and attacked by the same number, then the support and attack should cancel each other. This requirement is reasonable. Many valuations we know, including paper submissions to conferences and various impact factors used by universities in promotion considerations, involve numerical scores. It would be naive to assume that cancellation principles are not used, especially since many of the people involved may not be familiar with the subject matter.

So the beginning of this section, Section 4.1 tries to find the right functions for attack and support. We do find a good way of doing it and we discover to our surprise that the solution connects with the Dempster–Shafer rule (arising in a completely different community of researchers) and even more surprising is the connection with the cross-ratio of projective geometry. (The cross-ratio is used in geometry to define metrics on spaces.) This is investigated in Section 4.2.

By the end of Section 4.2, we will have drifted away from argumentation quite a bit. So the argumentation reader might ask why this should be of interest to him? We would say please expand your horizons, after years of research in logic, our considered opinion is that this is important.

This section discusses the addition of support arrows to argumentation networks. We will see that in order to have equal attack and support cancel each other, we need to reconsider the way we calculate the values of attacks (and supports). We offer a new definition and establish a connection between the new definition, the Dempster–Shafer rule, and surprisingly, the cross-ratio and projective metric distance from geometry.

4.1. *Discussion of support*

Consider a connection from a to b in Figure 35.

The double arrow indicates support. The simplest way to do it is to attack $(1 - y)$ which is the distance of b from 1.[16] Thus the new value of b is

$$\begin{aligned} 1 - (1-y)(1-\lambda x) &= 1 - [1 - \lambda x - y + \lambda xy] \\ &= \lambda x + y - \lambda xy = y + (1-y)\lambda x. \end{aligned}$$

If we have several supports, then $(1 - y)$ shrinks to

$$(1-y)(1-\lambda_1 x_1)(1-\lambda_2 x_2)\cdots(1-\lambda_k x_k)$$

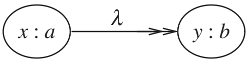


Figure 35. Numerical support.

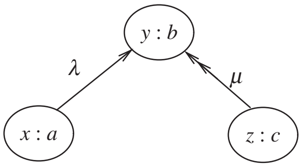


Figure 36. Both support and attack.

and the new value y' becomes $1 - (1 - y)\prod_i(1 - \lambda_i x_i)$. The difference $y' - y$ becomes

$$\begin{aligned}\Delta^+ y &= 1 - (1 - y)\prod_i(1 - \lambda_i x_i) - y\\ &= (1 - y)\left(1 - \prod_i(1 - \lambda_i x_i)\right)\end{aligned}$$

and we have

$$y' = y + \Delta^+ y.$$

How do we deal with both attack and support? Consider Figure 36, in this figure $x : a$ attacks $y : b$ and $z : c$ supports it. So the new value for b is

$$y - \lambda xy + \mu z(1 - y).$$

It is not clear what to do with several simultaneous attacks and supports. The model must be commutative in the order of application.

Our solution is simple. b is at a distance y from 0 and distance $1 - y$ from 1. Let the attackers attack y to get it nearer to 0 and let the supporters attack $(1 - y)$ to get b nearer to 1. Thus, if $x_i : a_i$ attack $y : b$ with transmission λ_i and $z_i : c_i$ support $y : b$ with transmission μ_i we get y' as the new value at b, where

$$\begin{aligned}y' &= y - \Delta^- y + \Delta^+ y\\ &= y - y\left(1 - \prod_i(1 - \lambda_i x_i)\right)\\ &\quad + (1 - y)\left(1 - \prod_i(1 - \mu_i z_i)\right)\\ &= y\prod_i(1 - \lambda_i x_i) + (1 - y)\left(1 - \prod_i(1 - \mu_i z_i)\right).\end{aligned}$$

Note that there is something numerically wrong with our proposal. In Figure 35, if we let $z = x$ and $\mu = \lambda$, i.e. the attack and support have the same values, then, we would have expected that they cancel each other. However, this is not the case. The new value is $y' = y - 2\lambda xy + \lambda x$.

This should not surprise us. The closer y is to 1, the less is the numerical value of an attack on $1 - y$, and the more numerical value we get for an attack on y. So, for example, assume $y = 0.9$ in value. Then a support of 50% of y will be half the distance of y from 1, i.e. will yield $\Delta^+ = 0.005$ in numerical value, while in comparison, a 50% attack on y will half the distance of y from 0 and will yield $\Delta^- = 0.45$. The net result of simultaneous attack and support will yield the new value $0.9 - 0.45 + 0.05 = 0.50$.

Can we remedy the situation? Perhaps,we should attack by changing the ratio $r(y)$ of y to $1 - y$, (i.e. $r(y) = y/(1 - y)$, and then calculate the new y' which will give the new ratio. So suppose the transmitted value (of attack or support) is $0 \leq \theta \leq 1$.

If θ is an attack we want to reduce $r(y)$ and so we let $r'(y) = \theta r(y)$. If θ is a support, we want to increase y, so the new ratio is $r'(y) = r(y)/\theta$.

We now solve the equation

$$\frac{y'}{1 - y'} = r'(y)$$

and therefore we get

$$y' = \frac{r'(y)}{1 + r'(y)}.$$

We must now decide on what value θ to use. Let us use the same value we used before, as agreed in Example 2.2. In Figure 35, we have $x : a$ attacking $y : b$ with transmission rate λ and we, therefore, have $\theta = (1 - \lambda x)$.

Let us calculate the values of attack and support with θ.

Case of attack

$$\begin{aligned} r'(y) &= \frac{y(1 - \lambda x)}{1 - y} \\ y' &= \frac{y(1 - \lambda x)}{(1 - y)(1 + y(1 - \lambda x)/(1 - y))} \\ &= \frac{y(1 - \lambda x)}{(1 - y + y - \lambda xy)} \\ &= \frac{y(1 - \lambda x)}{(1 - \lambda xy)} \end{aligned}$$

Case of support

$$\begin{aligned} r'(y) &= \frac{y}{(1 - y)(1 - \lambda x)}. \\ y' &= \frac{y}{(1 - y)(1 - \lambda x)(1 + y/(1 - y)(1 - \lambda x))} \\ y' &= \frac{y}{(1 - y)(1 - \lambda x) + y} \\ &= \frac{y}{(1 - \lambda x + y\lambda x)} \\ &= \frac{y}{1 - \lambda x + y\lambda x} \\ &= \frac{y}{1 - \lambda x(1 - y)} \end{aligned}$$

Let us now assume as before that the attack is 50%, e.g. $x = 0.5, \lambda = 1$.

We get $\theta = 0.5$. Assume as before $y = 0.9$. Hence $r'(y) = (0.9/0.1) \cdot 0.6 = 4.5$ and $y' = 4.5/(1 + 4.5) = 4.5/5.5 = 9/11$.

This should be compared with the previously attained value $0.45 = 9/20$.

For the support, we get

$$r'(y) = \frac{0.9}{0.1.0.5} = 18.$$

So

$$y' = \frac{18}{1+18} = \frac{18}{19}.$$

This should be compared with the value 0.05 we got previously.

How do we handle simultaneous attacks and supports? We follow the same principle as before. If $\theta_1, \ldots, \theta_n$ are attacking values and $\theta'_1, \ldots, \theta'_m$ are the supporting values then the new $r'(y)$ is

$$r'(y) = r(y)\frac{\prod_i \theta_i}{\prod_i \theta'_i}.$$

We shall argue below at the beginning of Section 4.2 that the choice of $\theta = (1 - \lambda x)$ is wrong. We can see that something is wrong qualitatively already. Consider Figure 36. We have that a attacks b and c supports b. So if a attacks b with $\theta = (1 - \lambda x)$ and c supports b with $\theta' = (1 - \mu z)$, then we get according to our formula that $r'(y) = r(y)\theta/\theta'$.

Now let us ask: does a attack c? And, does c attack a? What are the qualitative implications of these equations? Let us calculate. Assume for simplicity that $\lambda = \mu = 1$. We get no clear answers to our questions! However, if we were to take (for the case $\lambda = \mu = 1$) the value $\theta = (1 - x)/x$, and $\theta' = (1 - z)/z$ then we get that the sequence, a attacks c and then the already attacked c supports b, gives the same result as in Figure 36, which is why that a attacks c and b supports c in parallel. Notice that since $\theta = 1/r(z)$, we get the following rules, as summarised in Remark 4.1 below:

Remark 4.1 (Rules for attack and support) For x attacking y let $r'(y) = r(y)/r(x)$
For z supporting y let $r'(y) = r(y) \cdot r(z)$
For both, we get $r'(y) = r(y) \cdot r(z)/r(x)$
These equations imply that x attacks z and also z attacks x. The following three scenarios give the same results:

x attacks z and the modified z supports y;
z attacks x and the modified x attacks y;
x attacks y and z supports y in parallel.

Remark 4.2 (Loops involving support) Since we introduced support, we need to check what happens with loops involving support. For example, we need to check what happens with odd and even loops with support only.

In fact, in this section, we introduced a new algoirthm for numerical attack and support as summarised above in Remark 4.1.

So we need to check loops for this new type of numerical attack as well.

The first fact we observe, is that, by the definition of attack or support of $x : a$ on $y : b$, we use the ratios $r(x) = x/(1 - x)$ and $r(y) = y/(1 - y)$. For attack, we divide by the ratio to get the new ratio $r'(y) = r(y)/r(x)$ and for support, we multiply by the ratio to get the new ratio $r''(y) = r(y)r(x)$.

This means that $(x : a)$ attacking $(y : b)$ is the same as $((1 - x) : a)$ supporting $y : b$.

So when we have loops involving both attack and support, we can eliminate the supports by converting them into attacks.

Let us check Figure 37.

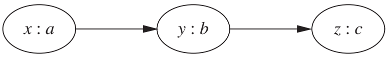


Figure 37. Converting support into attack in Figures 37 and 38.

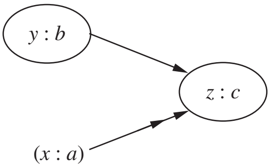


Figure 38. Converting support into attack in Figures 37 and 38.

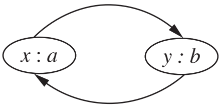


Figure 39. Typical numerical even loop.

From $x : a$ attacking $y : b$ we get a new ratio for b namely $r(y)/r(x)$.

Now the new ratio for b attacks $z : c$ and so the new ratio for c is

$$\frac{r(z)}{r(y)/r(x)} = \frac{r(z)r(x)}{r(y)}.$$

The new ratio for $z : c$ is the same as the ratio, we obtain if $y : b$ was attacking $z : c$ and $x : a$ was supporting $z : c$, as in Figure 38.

We get the new rate for c is $r(z) \cdot r(z)/r(y)$.

This is compatible with the intuition that if a attacks b which attacks c, then a actually supports c.

Let us now examine odd and even loops for the ratio type of attack. Consider Figure 39.

This is a typical even loop. Assume a steady-state solution. This means the following equations hold:

(1) Since b attacks a we must have

$$r(x) = \frac{r(x)}{(y)}.$$

(2) Since a attacks b we must have

$$r(y) = \frac{r(y)}{r(x)}.$$

From the equations, we get that $r(x) = r(y)$ and hence $r(x) = r(y) = 1$ and hence $x = y = \frac{1}{2}$.

This corresponds to Dung's argumentation theory to $x = y =$ undecided.

In Dung argumentation, we also have the solutions $x =$ in and $y =$ out and $x =$ out and $y =$ in.

These solutions correspond to the numerical solutions $(x = 0) \wedge (y = 1)$ and $(x = 1) \wedge (y = 0)$.

For $x = 0$ we have $r(x) = 1$ and for $x = 1$ we get $r(x) = \infty$.

Assume $f9(x) = 1$ ad $r(y) = \infty$, and check the equations. For the first equation we get

$$1 = \frac{1}{\infty} = 0$$

which is impossible.

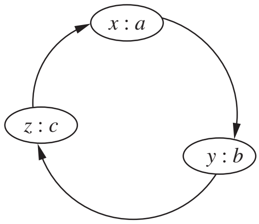


Figure 40. Numerical odd loop.

So we do not have these solutions in our case.

Let us now check the odd loop of Figure 40

In a steady-state solution, we get the following equations:

$$r(x) = r(x)/r(z)$$
$$r(y) = r(y)/r(x)$$
$$r(z) = r(z)/r(y)$$

The only solution is $r(x) = r(y) = r(z) = 1$, namely $x = y = z = \frac{1}{2}$.

In Dung argumentation terms, this corresponds to $a = b = c =$ undecided.

Now we ask what happens to these loops if some of the connections are supports? We saw that support of

$$(x : a) \twoheadrightarrow (y : b)$$

can be converted to attack

$$((1 - x) : a) \rightarrow (y : b).$$

The ratio $r(1 - x)$ equals the ratio of $1/r(x)$.

Now if we have a loop, odd or even, and some links are supports, we convert them into attacks with new ratio $= 1/$old ratio. Since the only solutions we got was that the ratios of all nodes is 1, then $1/$ratio is also 1 and this means we get all undecided as before.

We thus get that in our case:

- All loops, odd or even, no matter whether the links are attack or support, have only one solution: all undecided!

Remark 4.3 (Higher level attacks and support) The reader should note that our definition of numerical attacks and support as given in this section, really define how one number x can attack or support another number y. The number y' is calculated as summarised in Remark 4.1.

Therefore, when we consider higher level attacks or support of nodes on arrows or arrows on other arrows, such as we have in Figure 8, we have no problem calculating them. We just look at the numbers. So in Figure 8, for example, the node $z : c$ attacks the number ε representing the transmission rate of $(x : a) \rightarrow (y : b)$ with the transmission rate η. So the real attacking number is $z\eta$, attacking the number ε. The new number ε' is therefore

$$\varepsilon' = \frac{r(\varepsilon)/r(z\eta)}{1 + r(\varepsilon)/r(z\eta)} = \frac{r(\varepsilon)}{r(\varepsilon) + r(z\eta)}.$$

Remark 4.4 (Comparison with biological networks) It is worthwhile comparing the recursion results we obtained with the kind of recursion one gets in mathematical biology. We use the table (Table 3.1) of Turchin (2003, p. 53).[17]

(1) Old attack formula

$$y_{n+1} = y_n(1 - \lambda x).$$

This can be compared with exponential population growth.

(2) New attack formula

$$y_{n+1} = \frac{y_n(1 - \lambda x)}{1 - \lambda x y_n}.$$

This can be compared with the Beverton–Hort formula in the table of Turchin (2003, p. 53), see footnote 17.

Let us also examine what happens in the case of loops. Consider Figures 23 and 24 again. We have $v_1(e_1) = \lambda$ and the recursion equation, according to Figure 24 is

$$V_{n+1}(e_{n+1}) = \frac{\lambda(1 - \lambda V_n(e_n))}{1 - \lambda^2 V_n(e_n)}.$$

The recursion fixed point equation for this case is

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

or

$$V - \lambda^2 V^2 = \lambda - \lambda^2 V$$
$$\lambda^2 V^2 - \lambda^2 V - V + \lambda = 0$$
$$V^2 - \frac{(1 + \lambda^2)}{\lambda^2} V + \frac{1}{\lambda} = 0.$$

Let λ approach 1, we get

$$V^2 - 2V + 1 = 0$$

and so $V_\infty = 1$.

If we do the recursion proper, as in Figure 23, we get

$$V_{n+1} = \frac{V_n(1 - \lambda V_n)}{1 - \lambda V_n^2}.$$

The fixed point equation becomes

$$V(1 - \lambda V^2) = V(1 - \lambda V).$$

If we discard the solution $V = 0$, we get

$$1 - \lambda V^2 = 1 - \lambda V$$

hence

$$V^2 = V$$

and hence $V = 1$.

For more on biological networks, see Appendix 2.

4.2. *Connection with metric projective geometry and the Dempster–Shafer rule*

In the previous subsection, we agreed that in the situation of Figure 36, node a attacks node b by attacking the ratio:

$$r(y) = \frac{y}{1-y}.$$

We proposed that the attack value θ be $\theta = 1 - \lambda x$. We want to re-examine our decision in this subsection and see whether we can use a different attack value. First to simplify our qualitative consideration, assume $\lambda = 1$ and $\mu = 1$. Second, let us focus on node c, which is the supporting node b, with value z. Assume that z is very small, almost 0. One may feel that in many real applications, a very limited support is worse than nothing. It implies an attack on argument b, the hidden implication is that if b were any good why is not c's support of it a bit stronger? This way of thinking would integrate the support and attack together. So if a node supports another node with value z, then it simultaneously attacks it with value $1 - z$. If $z = 1$, then the support is complete. If $z \approx 0$ then the support is insulting and really accomplishes an attack to the value of $1 - z$.

Let us look at Figure 36 again. There are two ways to look at this figure (with $\lambda = \mu = 1$). One way is that we have two nodes, $x : a$ and $z : c$, the first attacking the node $y : b$ and the second supporting it.

The other way is that there is a single node $z : c$ supporting the node $y : b$, but simultaneously attacking it to the value $1 - z$, as discussed above. Figure 36, with $x = 1 - z$ is a representation of this new point of view through the additional node $x = (1 - z) : a$.

Of course, it is nicer to represent this new point of view directly, and indeed, later on in the paper we do represent this new point of view of support/attack mode by a double arrow.

Let us now calculate the new value y' of the attack and support configuration of Figure 36. We have:

$$r'(y) = \frac{y(1-x)}{(1-y)(1-z)}.$$

Hence

$$\begin{aligned} y &= \frac{r'(y)}{1 + r'(y)} \\ &= \frac{y(1-x)}{(1-y)(1-z) + y(1-x)} \\ &= \frac{y(1-x)}{1 - y - z + y(z + 1 - x)}. \end{aligned}$$

In order to compare with a later formula, let us rename the values. Let $z_2 = 1 - x$ and let $z_1 = z$. We get Equation (DS1):

$$y' = \frac{yz_2}{1 - y - z_1 + y(z_1 + z_2)}. \tag{DS1}$$

This equation means that a node $y : b$ is simultaneously supported by $z_1 : c$ and attacked by $(1 - z_2) : a$. Alternatively, we can say that the node is being [Support, Attacked] by the pair $[z_1, z_2]$. If $z_1 \leq z_2$ (i.e. $z + x \leq 1$), we can say we have a [Support, Attack] interval $[z_1, z_2], 0 \leq z_1 \leq z_2 \leq 1$.[18]

We adopt this terminology in preparation for the Dempster–Shafer point of view, yet to come, see item 3 of Example 4.5.

Let us now examine the case where $x = 1 - z$, i.e. $z = 1 - x$. We can view this as a [Support, Attack] pair $[z_1, z_2] = [z, z]$.[19]

We can view Figure 36 again and see that we are getting a situation of support value z from node c and attack value $1 - z$ from node a.

We have already calculated the new ratio $r'(y)$ for node b, it is

$$r'(y) = \frac{y(1-(1-z))}{(1-y)(1-z)} = \frac{yz}{(1-y)(1-z)}$$

Let us write this equation as

$$r'(y) = \frac{y/(1-y)}{(1-z)/z}. \tag{$*$}$$

We now calculate the new value y', it is

$$\begin{aligned} y' &= \frac{r'(y)}{1+r'(y)} \\ &= \frac{yz}{(1-y)(1-z)+yz}. \end{aligned} \tag{$**$}$$

We thus get that a node $z : c$ supporting a node $y : b$ yields the new value $y' : b$, where

$$y' = \frac{yz}{1-y-z+2yz} \quad \text{provided, of course, that } y+z-2yz \neq 1. \tag{DS2}$$

Let us say that $(*)$ and (DS2) represent a combined [Support, Attack] result of a node to a value $[z, z]$, attacking a node with value y.

We now connect (DS2) to the Dempster–Shafer rule (see Shafer 1976; Hajek et al. 1992), and to the cross-ratio and projective metric from geometry (see Faulkner 1949; Adler 1967).

Example 4.5 (Dempster–Shafer rule) The range of values we are dealing with is the set of all subintervals of the unit interval [0, 1]. The Dempster–Shafer addition on these intervals is defined by

$$[a,b] \oplus [c,d] = \left[\frac{a \cdot d + b \cdot c - a \cdot c}{1-k}, \quad \frac{b \cdot d}{1-k}\right],$$

where $k = a \cdot (1-d) + c \cdot (1-b)$, where '$\cdot$', '+', '$-$' are the usual arithmetical operations. The compatibility condition required on a, b, c, d is

$$\varphi([a,b],[c,d]) \equiv k \neq 1.$$

The operation $\oplus$ is commutative and associative. Let $\mathbf{e} = [0, 1]$.

The following also holds:

- $[a,b] \oplus \mathbf{e} = [a,b]$
- For $[a,b] \neq [1,1]$ we have $[a,b] \oplus [0,0] = [0,0]$
- For $[a,b] \neq [0,0]$ we have $[a,b] \oplus [1,1] = [1,1]$
- $[a,b] \oplus [c,d] = \varnothing$ iff either $[a,b] = [0,0]$ and $[c,d] = [1,1]$ or $[a,b] = [1,1]$ and $[c,d] = [0,0]$.

In this algebra, we understand the transmission value $[a, b]$ as saying that the actual transmission value lies in the interval $[a, b]$.

Let us make three comments as follows:

(1) Let x denote $[x, x]$. We get for $0 \leq a \leq 1$ and $0 \leq c \leq 1$ the following

$$\begin{aligned} a \oplus c &= \left[\frac{ac + ac - ac}{1 - a(1 - c) - c(1 - a}, \frac{ac}{1 - a(1 - c) - c(1 - a)}\right] \\ &= \left[\frac{ac}{1 - a - c + 2ac}, \frac{ac}{1 - a - c + 2ac}\right] \\ &= \frac{ac}{1 - a - c + 2ac} \end{aligned}$$

provided $(a + c - 2ac) \neq 1$.
We note immediately that (DS2) is $y \oplus z$. This is also the propagation method used by the MYCIN expert system (see Hajek and Valdes 1994).

(2) Let us check for what values of a, c can we have equality, i.e. when can we have $a + c - 1 = 2ac$?
Assume $a \leq c$.
We claim the only solution to the equation $a + c - 2ac = 1$ is $a = 0, c = 1$ for $a \leq c$ and $a = 1, c = 0$ for the case $c \leq a$. There is no solution for $c = a$.
To show this, let $c = a + \varepsilon, 0 \leq \varepsilon \leq c - a$.
Then assume

$$\begin{aligned} a + a + \varepsilon &= 1 + 2a(a + \varepsilon) \\ 2a + \varepsilon &= 1 + 2a^2 + 2\varepsilon a \\ \varepsilon - 2\varepsilon a &= 1 + 2a^2 = 2a \\ \varepsilon(\tfrac{1}{2} - a) &= a^2 - a + \tfrac{1}{2} \\ &= (a - \tfrac{1}{2})^2 - (\tfrac{1}{2})^2 + \tfrac{1}{2} \\ &= (a - \tfrac{1}{2})^2 + (\tfrac{1}{2})^2. \end{aligned}$$

Hence

$$\begin{aligned} \left(a - \frac{1}{2}\right)^2 + \varepsilon\left(a - \frac{1}{2}\right) + \left(\frac{1}{2}\right)^2 &= 0 \\ \left[\left(a - \frac{1}{2}\right) + \frac{\varepsilon}{2}\right]^2 - \left(\frac{\varepsilon}{2}\right)^2 + \left(\frac{1}{2}\right)^2 &= 0 \\ \left(a - \frac{1}{2} + \frac{\varepsilon}{2}\right)^2 &= \left(\frac{\varepsilon}{2}\right)^2 - \left(\frac{1}{2}\right)^2 \\ &= \left(\frac{\varepsilon}{2} - \frac{1}{2}\right)\left(\frac{\varepsilon}{2} + \frac{1}{2}\right). \end{aligned}$$

Hence $\varepsilon = 1$ and since $0 \leq c = a + \varepsilon \leq 1$ we must have $a = 0$ and $c = 1$.
In particular, we get that for $a = c = x$, $x \oplus x$ is always defined and we have

$$x \oplus x = \frac{x^2}{1 - 2x + 2x^2}.$$

For example, we have

$$0 \oplus 0 = 0$$
$$1 \oplus 1 = 1$$
$$\tfrac{1}{2} \oplus \tfrac{1}{2} = \tfrac{1}{2}$$

(3) Let us check what happens when $c = d$.
We get

$$\begin{aligned}[a, b] \oplus c &= \frac{bc}{1 - a(1-c) - c(1-b)} \\ &= \frac{bc}{1 - a + ac - c + bc} \\ &= \frac{bc}{1 - a - c + c(a+b)}.\end{aligned}$$

The reader should compare this equation with the formula (DS1) obtained before.

Example 4.6 (Cross-ratio) Consider the interval [0, 1] and two points y and $1 - z$ in this interval. Let $A = 0, B = 1, C = y$ and $D = 1 - z$. Taking AC, CB, AD, DB as directed intervals, we have it that $AC = y, CB = 1 - y, AD = 1 - z$ and $DB = z$.

The projective cross-ratio between these points, denoted traditionally by $(A, B; C, D)$ is calculated as the ratio of ratios of the directed intervals.

$$(A, B; C, D) = \frac{AC/CB}{AD/DB} = \frac{y/(1-y)}{(1-z)/z} = \frac{yz}{(1-y)(1-z)}.$$

Note that this is formula (*).

Note that this measures distance. In the Cayley–Klein metric, $\log(A, B; C, D)$ is used to describe the distance between points C and D. Figure 41 shows how it is done.

C and D are inside the unit circle. The chord connecting them meets the circle at A and B. See Adler (1967, Sections 4.10 and 11.7) and Faulkner (1949, Chapter 6).

Returning to Figure 36, we have

$$r'(y) = (0, 1; y, 1 - z). \tag{*}$$

We can now define a new kind of support/attack arrow (with value $z/1 - z$) in a network, as displayed in Figure 42 by double arrow

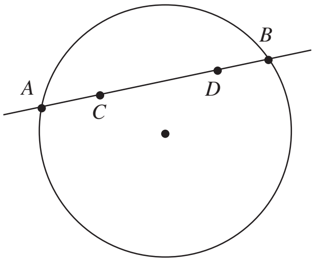


Figure 41. Cayley–Klein metric.

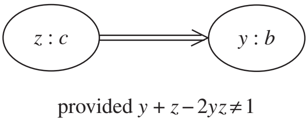


Figure 42. New kind of attack/support arrow.

We have for $0 \leq y, z \leq 1$

$$r(y) = \frac{y}{1-y}. \tag{$\sharp$1}$$

$$r'(y) = (0,1;y,1-z). \tag{$\sharp$2}$$

$$y' = \frac{yz}{1-y-z+2yz} = y \oplus z. \tag{$\sharp$3}$$

Furthermore, a formula (DS1) for a combined support to value z_1 and attack to value z_2, as in Figure 36, gives the result $y' = y \oplus [z_1, z_2]$. ($\sharp$4)

Equations ($\sharp$2)–($\sharp$4) open new opportunities for us.

(1) Allow for values to be intervals because of the connection with Dempster–Shafer.
(2) Allow for a connection with a more general non-Euclidean metric, using complex numbers.
(3) Attack and support values need not be in $[0, 1]$.

For further investigations, see Metcalfe, Olivetti, and Gabbay (2008).

Example 4.7 (Cross-ratio for intervals) This example will try to extend the notion of cross-ratio for intervals, i.e. we look for cross-ratio for $(0,1;[y_1,y_2],1-z), 0 \leq y_1 \leq y_2; 0 \leq z \leq 1$.

We saw that the situation in Figure 43 can be described as follows:

(1)

$$\begin{aligned} r(y,z) =& (0,1;y,1-z) \\ =& \frac{yz}{(1-y)(1-z)} \end{aligned}$$

(2) We also know that the Dempster–Shafer rule for the case of $y \oplus z = [y,y] \oplus [z,z]$ gives the value

$$y \oplus z = \frac{r}{1+r} = \frac{yz}{1-y-z+2yz}$$

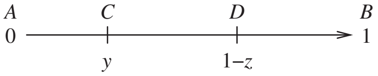


Figure 43. Cross Ratio.

(3) Our aim is to define cross-ratio $(0,1;[y_1,y_2],1-z)$. We use (2): Consider

$$[y_1,y_2]\oplus z=\frac{y_2z}{1-y_1-z+z(y_1+y_2)}$$

(4) Define by analogy with (2):

$$[y_1,y_2]\oplus z=\frac{r([y_1,y_2],z)}{1+r([y_1,y_2],z)} \qquad (*1)$$

we do not know what $r^*=r([y_1,y_2],z)$ means. However, using $(*1)$ and solving for r^* we get:

$$r^*=\frac{[y_1,y_2]\oplus z}{1-[y_1,y_2]\oplus z}.$$

Fortunately, the expressions on the right-hand side are all numbers: Hence we get

$$\begin{aligned}r^*&=\frac{y_2z}{1-y_1-z+z(y_1+y_2)(1-(y_2z/1-y_1-z+z(y_1+y_2)))}\\&=\frac{y_2z}{1-y_1-z+zy_1+zy_2-y_2z}\\&=\frac{y_2z}{1-y_1-z+zy_1}\\&=\frac{y_2z}{(1-y_1)(1-z)}\\&=\frac{y_2}{y_1}\cdot\frac{y_1z}{(1-y_1)(1-z)}=\frac{y_2}{y_1}r(y_1,z).\end{aligned}$$

We, therefore, have

$$r([y_1,y_2],z)=\frac{y_2}{y_1}r(y_1,z). \qquad (*2)$$

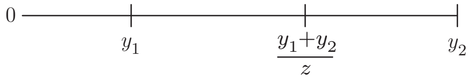
We can, therefore, define

$$(0,1;[y_1,y_2],1-z)=_{\text{def}}\frac{y_2}{y_1}(0,1;y_1,1-z) \qquad (*3)$$

or more generally:

$$(A,B;[C_1,C_2],D)=_{\text{def}}\frac{AC_2}{AC_1}(A,B;C_1,D). \qquad (\sharp)$$

Let us check whether $(\sharp)$ is invariant under some projective transformations.
Let us consider y_2/y_1. Think of it as a cross-ratio as in the figure below



$$\frac{(y_2-0)/(y_1-0)}{(y_2-(y_1+y_2)/2)/(y_1-(y_1+y_2)/2)}=\frac{y_1/y_1}{(y_2-y_1)/(y_1-y_2)}=\frac{-y_2}{y_1}$$

Thus

$$\frac{y_2}{y_1}=-\left(0,\frac{y_1+y_2}{2},y_1,y_2\right).$$

This cross-ratio uses the midpoint between y_1 and y_2. Midpoints E between points A and B can be characterised as the Harmonic conjugate of the point at infinity relative to A and B.

So any transformation of the line leaving the point at infinity fixed will also preserve midpoints, i.e. if A goes to A', B to B' and E to E' and ∞ to ∞, then if E is the midpoint of AB then E' is the midpoint of $A'B'$.

(5) Since $r(y,z)$ is commutative it stands to reason to define

$$r^{**} = r([y_1,y_2],[z_1,z_2]) = \text{def}\frac{y_2}{y_1}\cdot\frac{z_2}{z_1}r(y_1,z_1).$$

We now have a candidate definition for a cross-ratio for intervals.

$$r^{**} = \frac{y_2}{y_1}\frac{z_2}{z_1}\frac{y_1z_1}{(1-y_1)(1-z_1)}.$$

Hence

$$r^{**} = \frac{y_2z_2}{(1-y_1)(1-z_1)}. \tag{*3}$$

Let $\bar{y} = [y_1,y_2], \bar{z} = [z_1,z_2]$. Therefore, we can define a new $\boxplus$ using a similar connection as (*2):

$$\begin{aligned}\bar{y}\boxplus\bar{z} &= \frac{r^{**}}{1+r^{**}}\\ &= \frac{y_2z_2}{(1-y_1)(1-z_1)+y_2z_2}.\end{aligned}$$

Hence we summarise as follows:

$$\bar{y}\boxplus\bar{z} = \frac{y_2z_2}{1-y_1-z_1+z_1y_1+z_2y_2}. \tag{*4}$$

Let us compare $\boxplus$ with $\oplus$

We have

$$\begin{aligned}\bar{y}\oplus\bar{z} &= \left[\frac{y_1z_2+y_2z_1-y_1z_1}{1-y_1+y_1z_2-z_1+y_2z_1}, \frac{y_2z_2}{1-y_1+y_1z_2-z_1+y_2z_1}\right]\\ &= \left[\frac{y_1z_2+y_2z_1-y_1z_1}{1-y_1-z_1+y_1z_2+y_2z_1}, \frac{y_2z_2}{1-y_1-z_1+y_1z_2+y_2z_1}\right].\end{aligned}$$

They are not the same, unless $z_1 = z_2$ or $y_1 = y_2$.

To see this let us ask when do we get a point interval? We equate the numerators of the interval endpoint and we get

$$y_1z_2+y_2z_1-y_1z_1 = y_2z_2$$

hence

$$z_1(y_2-y_1) = z_2(y_2-y_1)$$

i.e. either $y_1 = y_2$ or $z_1 = z_2$ i.e. one has to be a point

Summary

We have extended the cross-ratio to a case of one interval, and it agrees with the Dempster–Shafer $\oplus$. We can also extend the cross-ratio to the case with two intervals, giving it the value

$$r^{**}(\bar{y},\bar{z}) = \frac{y_2z_2}{(1-y_1)(1-z_1)}$$

but it does not agree with the Dempster–Shafer $\bar{y}\oplus\bar{z}$.

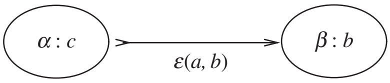


Figure 44. A new kind of arrow, indicating a general transmission.

We note, however, that since

$$r^{**}(\bar{y}, \bar{z}) = \frac{y_1}{y_1}, \frac{z_1}{z_1} r(y_1, z_1),$$

if we assume $y_1 = 1 - y_2, z_1 = 1 - z_2$, we get

$$r^*(\bar{y}, \bar{z}) = r(y_2, z_2) r(y_1, z_1).$$

We need to check what benefit this gives us!

Example 4.8 (Using Dempster–Shafer for attack and support) Consider again the basic situations depicted in Figures 36 and 42, or perhaps consider the more fundamental situation of Figure 7. Let us focus on the following Figure 44.

The new kind of arrow can stand in for attack, support or any combination transmitted from node c to node b. Our aim in this example is to review our options for the kind of values $\alpha, \beta, \varepsilon$ can take and the options available for the mathematical formulas for their combination and transmission.

Our previous discussion allows for the following Dempster–Shafer option

(1) $\varepsilon = 1, \alpha = [z_1, z_2], \beta = y, 0 \leq y \leq 1, 0 \leq z_1 \leq z_2 \leq 1$ and $y' = y \oplus [z_1, z_2]$ and the arrow is interpreted as the [Support, Attack] connection as in formula (DS1). We saw the connection with the cross-ratio as well.

(2) To maintain symmetry, we must also allow β to be of the form $[y_1, y_2], 0 \leq y_1 \leq y_2 \leq 1$ and we must write a formula for the [support, attack] on $\beta : b$. The obvious answer is to let

$$\beta' = \alpha \oplus \beta = [z_1, z_2] \oplus [y_1, y_2].$$

(3) Another possibility is to take $\boxplus$, i.e. let $\beta'' = \alpha \boxplus \beta$ (as in (*4) of the previous example) but then β'' is a number not a proper interval.

(4) Next let us ask what values can we give to ε? Again the simplest and most general value can be $\varepsilon = [u_1, u_2], 0 \leq u_1 \leq u_2 \leq 1$. We need to say how to combine it with α to get a value transmitted? Again in analogy with expert systems in AI, we can let the transmitted value to be $\alpha \oplus \varepsilon$. Thus, the new value β' would be

$$\beta' = \alpha \oplus \varepsilon \oplus \beta.$$

5. Numerical temporal dynamics overview

The discussion so far was static. The network is fixed and does not change with time. We discussed support and attack options and discussed loops but we did not discuss change. When we have change, we use the term *temporal dynamics* of networks. Let us begin.

5.1. *Introduction*

We devote this subsection to briefly outline some intuitive motivation for temporal dynamics. The reader should be aware that the temporal dynamic aspect of networks is central to the subject and will receive extensive study in future papers (see Barringer, Gabbay, and Woods 2012). Since our paper is on numerical networks, it is worthwhile to give a brief discussion of the special aspects of numerical change. We can use calculus in this case and involve rate of change in attacks and support.

We begin our discussion with general considerations of how to introduce time into a system. There are two ways that time can be introduced into the system. The (local) object level approach and the (global) meta-level approach. If the system is denoted by $\mathcal{S}$ and its components by $s_1, s_2, s_3, \ldots$ then the object level approach is to make each s_i time-dependent, i.e. $s_i = s_i(t)$. The global meta-level approach is to take temporal snapshots of the whole system at different times.

To illustrate the two approaches, consider a system of particles in mechanics. The local object level approach is to give a trajectory function $s(t)$ for each particle s. The meta approach is to take snapshots of the whole system at different times. Since the particles may be interacting, the meta approach is to give general differential equations governing the behaviour of the system while the object level approach gives the solutions to these equations, being the trajectory for each particle.

In the case of argumentation networks, both approaches are meaningful. The meta-level approach takes snapshots of the argumentation system at different times. Thus, we need an additional network of time points and we need to evaluate the argumentation network at the time network. We can use special connectives to express time behaviour and create new arguments involving time, out of the atomic arguments of the argument network. This is done in a continuation paper (Barringer et al. 2012) entitled Modal and Temporal Argumentation Networks.

For example, if a represents a proof for the existence of God, then Ha can represent the fact that (a new argument based on a) this proof has always been accepted as valid and thus one can use Ha as a new argument representing a stronger version of a.

The object level approach is to make the strength of a time-dependent. So a may represent a visual argument (by way of television footage) against involvement in some foreign war. The strength of this footage goes down as time moves on.

The two approaches may be combined. For example, one can argue that the strength $a(t)$ of acceptance of the proof of the existence of God has been going down over the years (as measured by per cents of the population who accept it), i.e. $\mathrm{d}a/\mathrm{d}t \ll 0$, and, therefore, we should no longer take such proofs into consideration!

So this argument has the form:

$$H\left(\frac{\mathrm{d}a}{\mathrm{d}t} \ll 0\right) \underset{\text{attack}}{\longrightarrow} a.$$

There is a third temporal aspect to networks and this is the time it takes to traverse (evaluate the arguments of) the network. This aspect is dominant in biological networks. We now elaborate more on this aspect:

Given an attack and support network, there are two interpretations for it which are intuitive and work well:

(1) biological interpretation;
(2) argumentation interpretation.

The biological network models population growth and dynamics varying in time. The network is fixed and time/population generations manifest themselves in propagating values in the network. Each complete cycle represents a generation in time.

In the case of argumentation networks, such cycles do not represent time, but the calculation of the strength of the various participating arguments. A stable solution of the cycling/propagation of the argumentation network gives the final value of the strength of the arguments.

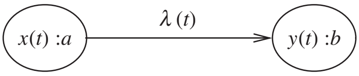


Figure 45. Attack with a time parameter.

In contrast, a stable solution of the propagation of attack and support values in the biological network represent as a steady-state population equilibrium.

An oscillatory solution in the biological case means population oscillation of various species, while in the argumentation case, such oscillatory behaviour is a problem because we want a steady answer to the values of the various nodes. Further machinery is needed in the argumentation case to rescue the situation and get a value for each node.

What would be a temporal aspect in the argumentation case?

Consider the simple network of Figure 45.

In this figure t is a time parameter. So the strength and transmission parameters of the net from a to b depend on time t.

The value of b is $y'(t) = y(t)(1 - \lambda(t)x(t))$.

We assume there that the transmission from a to b at any time is instantaneous. Thus, what we have is a parameterised family of networks, with parameter t.

There are several ways in which such a system can be made more interesting and more applicable.

(1) The variation of the network in time is continuous and has nice properties.
(2) Transmission takes time as opposed to transmission being instantaneous. By transmission being instantaneous, we mean that the process of Definition 2.9 takes no time at all.
(3) The variation is in some sense evolutionary, i.e. the value of the net at time $t + \Delta t$ is somehow dependent on the value at t. This dependence is governed by some evolutionary equations.

Let us examine one such simple case. Consider Figure 16 again.

Assume that at time $t = 0$, we have $x(0) = 1, \lambda(0) = 1$ and $y(0) = 1$. In this case $y'(0) = 0$.

So really argument b is totally defeated. However, if we know that the strength of $a(x(t))$ and its transmission rate $\lambda(t))$ decrease quickly, while the strength of $b, y(t)$ changes very slowly, then it is worth our while to wait a bit hoping the crisis (argument attack from a to b) will blow over.

Let us take for example

$$y(t) = \frac{1}{1+t}$$
$$x(t) = \frac{1}{1+te^t}$$
$$\lambda(t) = \frac{1}{1+te^t}$$

Hence for a small $t = \varepsilon$

$$y'(\varepsilon) = \frac{1 - 1/(1+\varepsilon e^{\varepsilon})^2}{1+\varepsilon}.$$

For $\varepsilon = 1$, we get $y(1) = \frac{1}{2}$ and in comparison, we have

$$\begin{aligned} y'(1) &= \frac{1 - 1/(1+e)^2}{2} \\ &= \frac{1}{2} - \frac{1}{2(1+e)^2} \\ &= \frac{1}{2} - 0.036. \end{aligned}$$

So there is a loss of about 7% as a result of the attack.

So if we are anxious to keep argument b, we might choose to wait a little (wait ε) for argument a and its transmission to weaken considerably.

Consider that we have

$a =$ sex scandal
$b =$ Governor to resign.

The chances are that public opinion will change quickly.

These time changes should be studied in the context of a time–action model. Suppose we have action **e** with precondition b and post-condition c. We want to take action **e** but if b is successfully attacked, we cannot do so. So we wait a bit. Conversely, suppose that we have d attacks a. Since d attacks a, b is available as $+b$ and so action **e** can be taken. But if d is weakening with time, we may choose to take action **e** immediately, while d is still 'saving' b by attacking a.

So a more sophisticated time–action–argument model will look at the speed of changes and will give values for actions to be taken.

We need to say more about what actions do in the model. We need to define the notion of a fact. We agree that syntactical facts e (as opposed to arguments), can be identified in our model by two properties:

(1) $V(e) = 1$
(2) e is not attacked by anything.

Of course there may be some arguments that have properties 1 and 2 above, but then for all practical purposes they are like facts.

There may be examples where it looks like some facts can be attacked by other facts. The fact that data is available on the computer may be attacked by the fact that a password was irretrievably lost. However, we can also look at the attack as focussing on the transmission rate of the fact and not the fact itself. We further accept that a node e is considered a semantical fact if $V(e) = 1$ and no attack arrows with positive transmission rate go into e. In a temporal dynamics model, these properties must hold at all times. If they hold only at some of the time, then e is not a fact but a commonly accepted truth which may sometime be attacked or doubted.

What do actions do? Actions create or destroy facts (see Gabbay and Woods 1998). So if at time t an action **e** is fired, then the result is that some facts get deleted from the network and some new facts are added. We can also assume that all values V change as a result of the action.

For simplicity, let us assume that an action adds only one fact or deletes only one fact. Since we can formally delete by attacking, we will only allow adding facts. By adding a fact, we mean either a new fact or turning an existing argument into a fact. So an action has the form **e** = (preconditions, post-conditions), where the precondition is a sequence of arguments $((x_i : a_i))$ and the post-condition is a sequence $((a \rightarrow y_i \rightarrow b_i))$. This means that we add the fact a and let it attack b_i with a transmission rate $y_i, i = 1, \ldots, n$. In a given network, if a is not a node then we add it as a node with value 1 and let it attack any node b_i which is in the network.

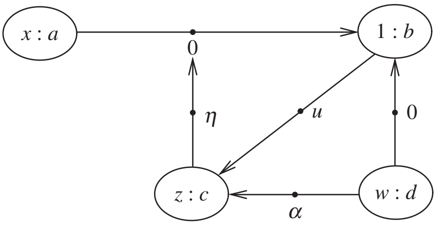


Figure 46. Applying an action to Figure 8 can yield this figure.

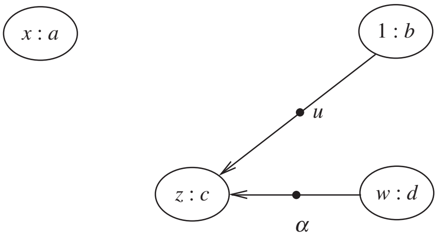


Figure 47. Figure 46 simplified.

If a is already in the network, then 'disconnect' all attacks on a by giving them value 0. Give a the value 1 and let a attack all existing b_i in the network. If b_i is already attacked by a with transmission rate u_i, then let the new transmission rate be $\max(u_i, y_i)$.

Note that **e** is stated independently of the network. To be activated, we need the net final value of a_i to be at least x_i and then the post-conditions act on the available b_i.

Example 5.1 Consider the network of Figure 8. Consider the action **e** with precondition $((x : a), (w : d))$ and post-condition $((b \to u \to c), (b \to y \to g))$. This action can be applied to the network of Figure 8.

The result is Figure 46 below. Note that since there is no node g in the network, $b \to y \to g$ is not implemented. This is equivalent to Figure 47.

The next question for us to answer in a temporal network is the following. If action **e** is activated at time t, when do we see the result? If the network operates in discrete time, then the result is at time $t + 1$. Otherwise, we have to treat the action like an *impulse* in a physical system, as when a ball hits another ball and gets it moving, and assume the result of the action **e** at t manifests itself immediately at all times s such that $t < s$. We have to give a reasonable definition of how the result of the action manifests itself. A good example for the initial consideration is that if a new argument e is created by an action at time t then it shows up at all times $s > t$ and its strength at time $s > t$ decays slowly as s increases, say it has the form $V_s(e) = k/(1 + s - t), k$ a constant ≤ 1. Similarly, we can ask for a decay of the transmission rates.

If we bring into the argumentation system the idea that traversing the network also takes time (as we have in the biological case), we get two time movements: the traversing time and the network change time.

Such a combination exists in legal arguments. Court cases take time to argue and 'evaluate' and thresh out the evidence and in parallel the laws and public perception of justice get changed. In many cases, they influence one another.

Let us go back to the biological case. Here the network does not change and the time components are the cycles (generations) through the network. So what can network change mean? This can

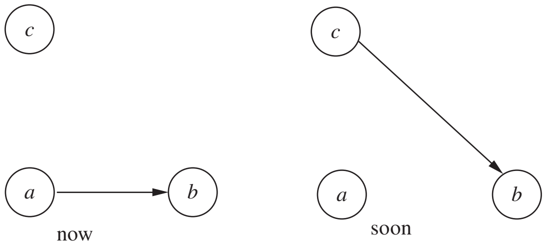


Figure 48. Time change example.

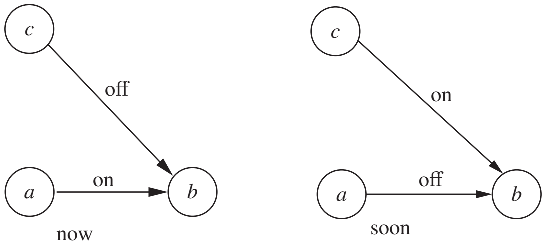


Figure 49. Another time change example.

be genetic mutations or genetic engineering which change the parameters in the system. The predator–prey relationship can change because of mutations in the prey, etc. Major disasters can affect the ecology. Deterioration of the habitat and environment can affect a slow continuous change in the parameters.

Some arguments lose their potency with the passage of time. This is well known in the political context. Politicians sometimes wait for the 'storm' to blow away, especially in matters of corruption and public protest. For example, some members of the UK Parliament (MPs) were recently exposed as claiming unjustifiably large expenses. There was a strong public protest to these findings, resulting in the resignation of some MPs. Many, however, have kept a low profile, awaiting for the public to forget. Let a represent the public protest over excessive expense claims, and b denote the standing of MPs, symbolically, we may have that now a attacks b, see Figure 48, but not for long, soon a will no longer attack b but a new attack on b, from c, say political in-fighting, may occur.

In such cases, we can represent the arguments and the attacks as time-dependent, $a(t), b(t)$ where t represents time. In contexts where arguments have strength (i.e. $a(t)$ is a number between 0 and 1), we can even consider the rate of change, $\mathrm{d}a/\mathrm{d}t$, $\mathrm{d}b/\mathrm{d}t$ and include it in our considerations. It may be convenient to represent the situation as in Figure 49. Where we do indicate attack arrows as on and off. Better still is to put labels on the attacks, e.g. $\mathbf{l}(c, b), \mathbf{l}(a, b)$, which can be on or off, and consider these labels as time-dependent.

6. Conclusion and discussion

We now conclude this paper. We embarked from traditional argumentation networks in the numerical direction. This allowed us to compare and see the place of traditional argumentation in the landscape of general networks, (which are mostly numerical). We concentrated our discussion on the handling of numerical loops and on the concepts of numerical attack and support. We discovered interesting connections and new ideas. What is left to be done in this section is a comparison with some related papers published since 2005 and a discussion of further research. This we do now.

6.1. *Comparison with related papers 2005–2011*

Our current paper is an expansion of our 2005 original paper (Volterra 1926). This paper was not always noticed by the argumentation community and so some related results have been published since 2005. We discuss some of these papers here. We are grateful to the referees for compiling this list of papers for us.

(1) The paper by Haenni (2009), on Probabilistic Argumentation: This paper relates to our paper in the following way. In our numerical argumentation networks, we have nodes of the form $(x : a)$ and attacks of the form $(x : a) \rightarrow (y : b)$ or support of the form $(x : a) \twoheadrightarrow (y : b), x$ and y are numbers in $[0, 1]$. We take these numbers as given, and do not ask where they come from. So suppose a is a fact and it is established because several witnesses with varying reliabilities testified to a. If we have a system of belief functions or probabilities on the sources of information that supports a, we can calculate a final number x to annotate a. The paper by Faulkner (1949) offers such a system. It is interesting to note that the Dempster–Shafer Rule does appear in this paper in the context of combining numbers and it also appears in our paper as well as we have seen in Section 4.2

(2) The paper by Leite and Martins (2011), on Social Abstract Argumentation: At the beginning of May 2011, J. Leite sent D. Gabbay a copy of this paper, which was submitted to *IJCAI*. He said he saw an abstract of Gabbay's (2011a) paper and thought it was related. Gabbay sent his papers (Gabbay 2011a, c). The paper by Leite and Martins (2011) is indeed related to the paper by Gabbay (2011c), which deals thoroughly with the equational approach to argumentation. In Gabbay and Rodrigues (2012), they addressed the ideas given in Leite and Martins (2011). The paper by Leite and Martins (2011) has in it implicitly as a side-effect a valuable principle relating to the computation of steady-state solutions to equations arising from numerical argumentation networks. It is a numerical analysis problem and is too elaborate to be discussed here. We shall address it in the expanded version of Gabbay and Rodrigues (2012).

(3)–(4) The paper by Cayrol and Lagasquie-Schiex (2005) on Graduality in Argumentation, and the paper by Matt and Toni (2008) on A Game-Theoretic Measure of Argument Strength for Abstract Argumentation: These two papers are concerned with the following problem: ordinary Dung argumentation and the Caminada labelling classify the arguments into three classes only: *in*, *out* and *undecided*. These authors share the view that a finer classification is needed. Thus, if we look at Figure 3, we see that a is not attacked by anything, while c is attacked by b and defended by a. There is a unique extension, grounded extension, $a = \text{in}$, $b = \text{out}$, $c = \text{in}$.

The view of these authors is that a is much more valued in, while c is in, but not so much valued as a.

So the more attacked a node x is, the less value it has, even if it is in.

We, therefore, seek to define a numerical valuation function $\lambda x V(x)$ on arguments to reflect this situation. Cayrol and Lagasquie-Schiex (2005) use the geometry of the network to define such a function, $V(x)$, by looking at chains of attack and defence leading to the element x. Matt and Toni (2008) also use the geometry of the network, looking at the attackers of x and at the arguments which x attacks and by means of the game theoretical methods gives a numerical value $V(x)$.

Our analysis of these papers is as follows:

(a) First they offer a numerical valuation of points $x \in S$ in an argumentation network (S, R), which reflect their geometric-topological position in the network.

(b) Second they claim that this numerical value can be used to give some meaningful value as to how ‘in’ or ‘out’ the element x is.

Giving values to points in (S, R) is purely mathematical and is a well-established practice. This is how metrisation theorems in topology are proved and this is how metrics are introduced in projective geometry, by means of the cross-ratio. However, connecting these values obtained with arguments and saying that one argument is better than the other based on these numerical values is a different matter. One may not subscribe to the basic idea that an argument not attacked at all is more ‘in’ than an argument which is attacked and defended.

There are also technical reasons against this view. If (S_1, R_1) is a subsystem of (S_2, R_2), the relative valuations of any two poitns $s, y \in S_1$ may change, depending on whether we view them as part of S_1 or part of S_2, giving the feeling that arguments have no individual merit in themselves, but depend only on the context in which they are used.

Now let us compare the systems of Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008) with our system in this paper. When we write $(x : a)$, giving value x to argument a, the value x is external, calculated by a method outside the network, like how reliable a is, or x is a value obtained by probability on some evidence as in Haenni (2009), etc. Since these values are obtained or given externally, we need to say how to calculate transmitted values following attacks and support. So if $(x : a)$ attacks $(y : b)$, we need to say what is the new value y' of b, following the execution of the attack or support.

The values obtained by the geometrical methods of Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008), already take into consideration the geometry of the network. They are, therefore, final values. We cannot apply our machinery to them.

On the other hand, we can apply geometric machinery to networks with numerical values, considering for geometrical purposes the items in the network as slightly more complex, namely of the form $(x : a)$. So the values obtained by Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008) have a different standing altogether, than what we do.

Consider for example, a node a in the network which is being attacked by other nodes. In our system, we can give it a value $x = 1$ (i.e. we allow $(1 : a)$). In the systems of Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008), the value 1 is reserved (we think) only to points in the network with no geometrical attackers.

Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008) can be compared, however, with the equational approach of Gabbay (2011a,c). The equations are written based on the geometry of the network and the solution to the equations give numerical values to nodes. This can be seen as a third method of making distinctions between nodes which also makes use of the geometry of the network.

(5) The paper by Cayrol, Devred and Lasasquie-Schiex (2010), on acceptability semantics accounting for strength of attacks in argumentation.

This paper considers a finite number of types of attacks, organised in order of strength. We can present the network as say $(S, R_1, \ldots, R_n)$ with $R_i \subseteq S^2$ being pairwise disjoint, and with R_i being considered stronger than $R_{i=1}, 1 \leq i < n$. Consider, for example, Figure 50:

In this figure c is attacked by b but is defended by a. We would expect the extension $\{a, c\}$. However, we get this extension only if $k \geq l$.

Comparing this paper with our paper, there are two ways to look at it:

(a) Since the number of types of attacks is finite, we can look at the paper as a contribution in the direction of classical logic argumentation either in the spirit of

Gabbay and Szalas (2009) or in the spirit of Gabbay's (2011b). The framework is classical logic, there are several types of attacks $R_1, \ldots, R_n$ and they are used in one way or another to define extension. The extra requirement that R_i are disjoint and that there is a meta-level ordering on $\{R_i\}$, is just one possible axiomatic restriction on the system (though these are exactly the axioms which allow us to talk about the strength of attack).

(b) The second point of view is to say the system of Cayrol and Lagasquie-Schiex (2005) is indeed a case of strength of attacks which is a special case of our system. We can simulate the system of Cayrol and Lagasquie-Schiex (2005) in our system as a special case. It is not our purpose here to write a mini paper about Cayrol and Lagasquie-Schiex (2005), but simply to compare and give the reader an orientation. To this end, consider Remark 4.2 and Figure 37 and compare it with Figure 50. They are very similar. Now consider Figure 38. This figure was shown equivalent in our system (as discussed in Remark 4.2) to Figure 37. Therefore, we can look now at the 'corresponding' Figure 51. In this figure b attacks c with strength l while a supports c with strength k. Clearly in this picture c will survive only if $k \geq l$.

The above is just one possibility of how we could simulate Cayrol and Lagasquie-Schiex (2005) as part of our system.

(6)–(12) Papers on higher level attacks, also called AFRAS and Baroni et al. (2009a, 2011), Gabbay (2009a), Modgil (2009), Modgil and Bench-Capon (2008, 2010), and Crochemore and Gabbay (2011): These papers have to do with higher level attacks. An argument x can either attack another argument u, written $x \rightarrow u$, or attack the attack of a second argument y attacking a third argument z, written $x \twoheadrightarrow (y \rightarrow z)$. The idea of nodes (arguments, possible worlds in modal logic) attacking other arrows (attacks in argumentation, accessibility arrows in modal logic) originates in Gabbay (2004, 2008) in the context of Reactive Kripke Semantics. Such semantics is strictly stronger than traditional Kripke semantics and turned out to have a wide range of applications, not only in the usual application areas of modal logic, but also in many other areas such as automata theory (Crochemore and Gabbay 2011) and argumentation. In our 2005 paper (Barringer et al. 2005), we imported this notion to general attack networks and in Gabbay and Woods (2003b), we used this notion in argumentation and legal reasoning. Baroni et al. (2009a, 2011), Modgil (2009), Modgil and Bench- Capon (2008), Modgil and Bench-Capon (2010), independently introduced attacks on attacks. Modgil and

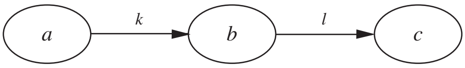


Figure 50. Figure for comparison in (5b).

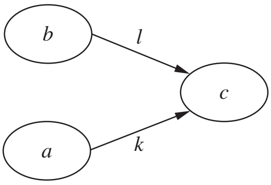


Figure 51. Figure for comparison in (5b).

Bench-Capon considered the form $x \twoheadrightarrow (y \rightarrow z)$ and Baroni, Cerutti, Giacomin and Guida, considered recursive higher level attacks of the form $x \twoheadrightarrow$ (higher level arrow). No one considered attacks of the form arrow $\twoheadrightarrow$ arrow, the kind we have in Barringer et al. (2005) and in this paper.

The problem with such higher level attacks is how to define the semantics and extensions. For the latest on this problem, see Gabbay (2009a), Baroni et al. (2011), and Hanh, Dung, and Thang (2011), as well as a section in Gabbay (2011c) presenting the equational approach to higher level attacks.

(13)–(14) Support in argumentation networks, BAFs: Our current paper, as well as Barringer et al. (2005), deals with support in the numerical context. We focussed on the qualitative idea that if there is support with numerical value x and attack with numerical value x, they nullify one another. This makes sense only in the numerical context. Support in non-numerical context is a very hot area, still in a state of flux. See the survey in a handbook chapter by Cayrol and Lagasquie-Schiex (2009). We are also working on it, (Boella, Gabbay, van der Torre, and Villata 2010), but the area is still evolving.

6.2. *Further research*

Let us summarise in this concluding section the new ideas emerging in this paper, which show great potential and require further research.

(1) *Neuro-fuzzy argumentation networks*
This is a natural generalisation of the numerical strength idea of Sections 1 and 2. It has not been picked up in the argumentation community and not discussed in this paper. We are going to pursue it extensively in a planned forthcoming book.

(2) *Higher level attacks*
These are the reactive arrows emanating from arrows to arrows. The community have rediscovered this both as a concept and as a tool. In Gabbay (2009a), we show how to reduce this concept to the object level by adding more arguments. In fact,we have a forthcoming book, (Gabbay 2012a), dealing with the meta-level approach to argumentation.

(3) *Connection with cross-ratio of projective geometry*
We believe this is very promising as well as mysterious. We still have to fully understand and exploit this connection. The key figure here is the situation described in Figure 41. C is attacking D and the result is calculated using the cross-ratio with A and B. This is pure geometry on the line AB. So C and D can be vectors of length n, each containing multiple context incomparable arguments attacking the corresponding componentwise arguments. What would be the result? If we regard C and D as vectors in n-dimensional space and pass the line CD through the n-dimensional unit sphere we get the points A and B and we can find the result. It is the point E on the line AB at a distance e from the point A where $e/(1-e) =$ The cross-ratio$(A, B; C, D)$
This locates the point E which we can take as the result of C attacking D in n-dimensional space.

(4) *Temporal dynamics*
See our papers (Barringer and Gabbay 2010; Barringer et al. 2012) for non-numerical temporal argumentation.

(5) *Geometrical numerical argumentation networks*

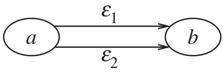
We were impressed with papers by Cayrol and Lagasquie-Schiex (2005) and Matt and Toni (2008) that were discussed in Section 6.1. We would like to offer our own views in the matter which include both attack and support arguments.

Acknowledgements

We are grateful to the referees for a very thorough review and penetrating comments. Research done under Israel Science Foundation project 1321/10: Integrating Logic and Network Reasoning.

Notes

1. Note that the concept of attack on attack is geometrical and applies to any network and so priority really lies with (Barringer et al. 2005).
2. Attacking the character of a witness presenting an argument may or may not be relevant to the argument. For example, attacks on the private life of an expert witness are unlikely to be relevant to the credibility of her expertise and hence her arguments, whereas attacks on the credentials of the expert witness's expertise are more likely to be relevant.
3. The model also allows several attacks to emanate from the same argument, as in the figure below.



The idea here is that there are several different kinds of arguments as to why a is an attack upon b. This makes sense, especially if a is a fact (see below). Such formal networks exist in the literature as *transition systems*, and the different arrows from a to b represent different actions, leading from state a to state b.

4. We mention that the idea of higher level attacks is none other than the idea of reactive arrows in networks and Kripke models, originating in Gabbay (2004), and applied to argumentation networks in Barringer et al. (2005).
5. In general, the value transmitted should be a function $\tau(\varepsilon, x)$ of ε and x, monotonically increasing in ε and x, with $\tau(0,0) = 0, \tau(1,1) = 1$. We choose multiplication here by way of example.
6. In Bayesian nets there are no $\varepsilon_1, \ldots, \varepsilon_n$. x_i are the probabilities associated with the nodes a_i and $\mathbf{f}$ is the conditional probability of node b relative to all the a_i. Thus, the probability y of b can be calculated.
7. When the arc is an attack, we use the curly arrow to represent it, instead of a traditional straight one-headed arrow $\rightarrow$. When the arc is a support (see Section 4) we use the double arrow $\twoheadrightarrow$.
8. For example, if z is a supporter of a theory of love and peace, then any x attacking y is in itself an attack on z. Also in those applications where it makes sense to traverse the network along its edges, then traversing the edge from x to y may trigger an attack on z. The formal machinery of representing such attacks exists in Gabbay (2009a, c).
9. Compare with Definition 1.1 of Gabbay (2009a), which deals with ordinary argumentation networks with $\{0, 1\}$ values.
10. By restricting $\mathbf{f}$ to finite sequences, we are forced to impose the condition of finitely branching on T in Definition 2.9 below. However, $\mathbf{f}$ can be more general, for example, we can take

$$\mathbf{f}'(y, S) = \inf\{\mathbf{f}(y, \bar{x}, \bar{\varepsilon}) \mid (\bar{x}, \bar{\varepsilon}) \in S\},$$

where S can now be an infinite set. This will allow us more freedom in Definition A.1 below.

11. When we say $\mathbf{f}$ is symmetrical, we mean that for any permutation σ of $\{1, \ldots, n\}$, we have

$$\mathbf{f}(y, \bar{x}_i, \bar{\varepsilon}_i) = \mathbf{f}(y, \bar{x}_{\sigma(i)}, \bar{\varepsilon}_{\sigma(i)}).$$

12. The notion of semantic acyclicity is discussed at the end of the section.
13. Consider a linear network of n nodes with the following connection structure. Label the nodes from 1 to n. Node i attacks all nodes numbered $> i$. Wave 0 will have to search n nodes. Wave $i < n$ will have to search $n - i$ nodes. The sum of all waves is $n(n-1)/2$.

14. Note that if we solve the fixed point recursion equation $V_{\lambda,\infty} = \lambda(1 - \lambda V_{\lambda,\infty})$, we get $V_{\lambda,\infty} = \lambda/(1 + \lambda^2)$.
15. The fixed point recursion equation for this case is $V = V(1 - \lambda V)$, yielding $V = 0$. We get the same equation if we follow option (d) for Figure 23.
16. This is Bernouli's rule of combination (see Shafer 1976, pp. 75–76).
17. We quote: Some functional forms proposed for single-species discrete-time models of population growth ($\lambda_0 = \exp[r_0]$ is the intrinsic discrete or multiplicative rate of population increase, k is the carrying capacity, b some positive constant, and θ an exponent)

Label	*Function*
Exponential	$N_{t+1} = \lambda_0 N_t$
Quadratic map	$N_{t+1} = \lambda_0 N_t(1 - N_t/k)$
Ricker	$N_{t+1} = \lambda_0 N_t \exp[-bN_t]$
Gompertz	$N_{t+1} = \lambda_0 N_t^{\theta}$
Beverton–Holt	$N_{t+1} = \lambda_0 N_t/(1 + bN_t)$
Desensation	$N_{t+1} = \lambda_0 N_t^2/(1 + bN_t^2)$
Theta–Ricker	$N_{t+1} = \lambda_0 N_t \exp[-bN_t^{\theta}]$

18. Actually the intervals involved are $[0, z_1], [1 - z_2, 1]$.
19. Beware of some possible confusion in notation. In Figure 36, the attack of a node is with value $x = 1 - z$ and the support is with value z. If we regard Figure 36 as representing the [Support, Attack] double arrow of Figure 42, we write it as $[z, 1 - x] = [z, z]$ and *not as* $[z, x]$. This is because $z_2 = (1 - x)$ appears in (DS1).

References

Abraham, M., Gabbay, D., and Schild, U. (2011a), *Resolution of Conflicts and Normative Loops in the Talmud*, College Publications, London UK.

Abraham, M., Gabbay, D.M., and Schild, U. (2011b), 'The Handling of Loops in Talmudic Logic, with Application to Odd and Even Loops in Argumentation', in *Proceedings of Howard 60*, eds. D. Rydeheard, A. Voronkov, and M. Korovina, Report, Department of computer science, University of Manchester, Manchester UK, pp. 1–25, earlier version published in 2011 as part of the monograph.

Adler, C. (1967), *Modern Geometry* (2nd ed.), McGraw Hill, New York and London.

Anderson, R.M., Turner, B.D., and Taylor, L.R. (eds.) (1979), *Population Dynamics*, Blackwell, Oxford UK.

Augusto, J.C., and Simari, G.R. (1999), 'A Temporal Argumentative System', in *AI Communications archive* (AI-99) IOS Press Amsterdam, The Netherlands, The Netherlands 12, 237–257.

Baroni, P., and Giacomin, M. (2003), 'Solving Semantic Problems with Odd-Length Cycles in Argumentation', in *Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003)*, LNAI 2711, Aalborg, Denmark: Springer-Verlag, pp. 440–451.

Baroni, P., Giacomin, M., and Guida, G. (2005), 'SCC-Recursiveness: A General Schema for Argumentation Semantics', *Artificial Intelligence*, 168(1–2), 162–210.

Baroni, P., Cerutti, F., Giacomin, M., and Guida, G. (2009a), 'Encompassing Attacks to Attacks in Abstract Argumentation Frameworks', in *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 83–94.

Baroni, P., Cerutti, F., Giacomin, M., and Guida, G. (2009b), 'An Argumentation-Based Approach to Modeling Decision Support Contexts with What-If Capabilities', in *2009 AAAI Fall Symposium Series: The Uses of Computational Argumentation.*

Baroni, P., Cerutti, F., Giacomin, M., and Guida, G. (2011), 'Afra: Argumentation Framework with Recursive Attacks', *International Journal of Approximate Reasoning*, 52(1), 19–37.

Barringer, H., and Gabbay, D. (2010), 'Modal and Temporal Argumentation Networks', in *Amir Pnueli Memorial Volume: Time for Verification*, eds. D. Peled and Z. Manna, Lecture Notes in Computer Science, Vol. 6200, Springer, Berlin Heidelberg, pp. 1–25.

Barringer, H., Gabbay, D., and Woods, J. (2005), ‘Temporal Dynamics of Support and Attack Networks’, in *Mechanising Mathematical Reasoning*, eds. D. Hutter and W. Stephan, Lecture Notes in Computer Science, Vol. 2605, Springer, Berlin Heidelberg, pp. 59–98.

Barringer, H., Gabbay, D., and Woods, J. (2008), ‘Network Modalities’, in *Linguistics, Computer Science and Language Processing, Festschrift for Franz Guenthner on the Occasion of his 60th Birthday*, eds. G. Gross and K.U. Schulz, College Publications, London UK pp. 70–102.

Barringer, H., Gabbay, D., and Woods, J. (2012), ‘Modal and Temporal Argumentation Networks’, *Argumentation and Computation*, Special issue, 3(2–3).

Bench-Capon, T. (2003), ‘Persuasion in Practical Argument Using Value Based Argumentation Framework’, *Journal of Logic and Computation*, 13, 429–448.

Besnard, P., and Hunter, A. (2001), ‘A Logic-Based Theory of Deductive Arguments’, *Articial Intelligence*, 128(1–2), 203–235.

Bodanza, G.A., and Tohmé, F.A. (2009), ‘Two Approaches to the Problems of Self-Attacking Arguments and General Odd-Length Cycles of Attack’, *Journal of Applied Logic*, 7, 403–420.

Boella, G., Kaci, S., and van der Torre, L. (2009), ‘Dynamics in Argumentation with Single Extensions: Attack Refinement and the Grounded Extension (Extended Version)’, *ArgMAS 2009*, 150–159. Argumentation in Multi-Agent Systems Lecture Notes in Computer Science Volume 6057, 2010, pp. 150–159.

Boella, G., Gabbay, D.M., van der Torre, L., and Villata, S. (2010), ‘Support in Abstract Argumentation’, in *Computational Models of Argument*, *COMMA 2010*, eds. P. Baroni, F. Cerutti, M. Giacomi, and G. Simari, IOS Press, Amsterdam, pp. 111–122, expanded version to appear in *AMAI*, 2012, special issue.

Caminada, M.W.A. (2006), ‘On the Issue of Reinstatement in Argumentation’, *JELIA*, pp. 111–123. http://icr.uni.lu/∼martinc/publications/JELIA_reinstatement.pdf

Caminada, M.W.A. (2007), ‘An Algorithm for Computing Semi-Stable Semantics’, *ECSQARU 2007*, pp. 222–234. http://icr.uni.lu/∼martinc/publications/algorithm_eCSQARU.pdf

Caminada, M.W.A. (2011), ‘A Labelling Approach for Ideal and Stage Semantics’, *Argument and Computation*, 2(1), 1–21.

Caminada, M.W.A., and Amgoud, L. (2007), ‘On the Evaluation of Argumentation Formalisms’, *Artificial Intelligence*, 171(5–6), 286–310.

Caminada, M., and Gabbay, D.M. (2009), ‘A Logical Account of Formal Argumentation’, *Studia Logica*, 93(2–3), 109–145.

Cayrol, C., and Lagasquie-Schiex, M.-C. (2005), ‘Graduality in Argumentation’, *Journal of Artificial Intelligence Research (JAIR)*, 23, 245–297.

Cayrol, C., and Lagasquie-Schiex, M.-C. (2009), ‘Bipolar Abstract Argumentation Systems’, in *Argumentation in Artificial Intelligence*, eds. G. Rahwan and R. Simari, Springer, Berlin Heidelberg pp. 65–84.

Cayrol, C., Devred, C., and Lagasquie-Schiex, M.-C. (2010), ‘Acceptability Semantics Accounting for Strength of Attacks in Argumentation’, in *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence* IOS Press Amsterdam, The Netherlands, The Netherlands ©2010 pp. 995–996 [long version available at ftp://ftp.irit.fr/IRIT/ADRIA/rap-2010-13.pdf]

Cobo, M.L., Martinez, D.C., and Ricardo Simari, G. (2010), ‘On Admissibility in Timed Abstract Argumentation Frameworks’, in *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence* IOS Press Amsterdam, The Netherlands, The Netherlands ©2010, pp. 1007–1008.

Crochemore, M., and Gabbay, D. (2011), ‘Reactive Automata’, *Information and Computation*, 209(4), 692–704, published on line DOI: 10.1016/j.ic.2011.01.002.

d’Avila Garcez, A.S., Gabbay, D.M., and Lamb, L.C. (2004), ‘Argumentation Neural Networks’, in *Proceedings of 11th International Conference on Neural Information Processing (ICONIP’04)*, Calcutta, India, Lecture Notes in Computer Science, Springer-Verlag.

d’Avila Garcez, A.S., Lamb, L.C., and Gabbay, D. (2008), *Connectionist Non-classical Logics: Distributed Reasoning & Learning in Neural Networks* (Monograph). Springer-Verlag. Berlin Heidelberg.

D’Inverno, R. (1992) *Introducing Einstein’s Relativiety*, Oxford University Press, Oxford UK.

Dung, P.M. (1995), ‘On the Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning, Logic Programming and *N*-Person Games’, *Artificial Intelligence*, 77, 321–357.

Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., and Wooldridge, M. (2009), 'Inconsistency Tolerance in Weighted Argument Systems', in *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 851–858.

Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., and Wooldridge, M. (2011), 'Weighted Argument Systems: Basic Definitions, Algorithms and Complexity Results', *Artifical Intelligence* 175(2), 457–486.

Faulkner, T. (1949), *Projective Geometry*, Edinburgh UK, Oliver and Boyd.

Gabbay, D.M. (1996), *Fibring Logics*, Oxford University Press.

Gabbay, D.M. (2002), 'Theory of Hypermodal Logics', *Journal of Philosophical Logic*, 31, 211–243.

Gabbay, D. (2004), 'Reactive Kripke Semantics and Arc Accessibility', in *Proceedings of CombLog04* (http://www.cs.manth.ist.utl.pt/comblog04), eds. W. Carnielli, F.M. Dionesio, and P. Mateus, Centre of Logic and Computation, University of Lisbon, pp. 7–20. ftp://logica.cle.unicamp.br/pub/e-prints/comblog04/gabbay.pdf.

Gabbay, D. (2008), 'Reactive Kripke Semantics and Arc Accessibility', in *Pillars of Computer Science: Essays dedicated to Boris(Boaz) Trakhtenbrot on the Occasion of his 85th Birthday*, eds. A. Avron, N. Dershowitz, and A. Rabinovich, Lecture Notes in Computer Science 4800, Springer, Berlin Heidelberg, pp. 292–341.

Gabbay, D. (2009a), 'Semantics for Higher Level Attacks in Extended Argumentation Frames, Part 1: Overview', *Studia Logica*, 93, 355–379.

Gabbay, D.M. (2009b), 'Modal Foundations for Argumentation Networks', *Studia Logica*, 93(2–3), 199–230.

Gabbay, D.M. (2009c), 'Fibring Argumentation Frames', *Studia Logica*, 93(2–3), 231–295.

Gabbay, D.M. (2011a), 'Introducing Equational Semantics for Argumentation Networks, in *Proc. ECSQARU 2011*, ed. W. Liu, LNAI 6717, Springer-Verlag, Berlin Heidelberg, pp. 19–35, doi: 10.1007/978-3-642-22152-1 2.

Gabbay, D. (2011b), 'Dung's Argumentation is Equivalent to Classical Propositional Logic with the Peirce-Quine Dagger', *Logica Universalis*, 5(2), 255–318, doi: 10.1007/s11787-011-0036-3.

Gabbay, D.M. (2011c), 'An Equational Approach to Argumentation Networks', *Argumentation and Computation*, Special Issue, 3(2–3).

Gabbay, D. (2012a), *Meta-Logical Investigations in Argumentation Networks*, forthcoming book, Springer. Berlin Heidelberg, 500 pp.

Gabbay, D. (2012b), 'The Equational Approach to CF2 Semantics', *Proceedings COMMA 2012*, IOS Press, short version to appear.

Gabbay, D.M., and d'Avila Garcez, A.S. (2009), 'Logical Modes of Attack in Argumentation Networks', *Studia Logica*, 93(2–3), 199–230.

Gabbay, D.M., and Rodrigues, O. (2012), A numerical approach to the merging of argumentation networks, To appear, Proceedings of Computational Logic in Multi-Agent Systems, CLIMA 2012, editors M Fisher and L van der Torre , SPRINGER VERLAG Berlin Heidelberg, 2012.

Gabbay, D.M., and Szalas, A. (2009), 'Annotation Theories over Finite Graphs', *Studia Logica*, 93, 147–180.

Gabbay, D.M., and Woods, J. (1998), '*Ad Baculum* is not a Fallacy', in *Proceedings of the Fourth Internatioal Conference of the International Society for the Study of Arguementation*, eds. F.H. van Eemeren, R. Grootendorst, J.A. Blair, and C.A. Willard, Amsterdam: SicSat, pp. 221–224.

Gabbay, D.M., and Woods, J. (2001a), 'Non-Cooperation in Dialogue Logic', *Synthese*, 127, 161–180.

Gabbay, D.M., and Woods, J. (2001b), 'More on Non-Cooperation in Dialogue Logic', *Logic Journal of the IGPL*, 9, 305–324.

Gabbay, D.M., and Woods, J. (2002), 'Formal Approaches to Practical Reasoning', in *Handbook of the Logic of Argument and Inference: The Turn Towards the Practical*, eds. D.M. Gabbay, R.H. Johnson, H.J. Ohlbach, and J. Woods, Amsterdam: North-Holland, pp. 449–481.

Gabbay, D.M., and Woods, J. (2003a), 'Normative Models of Rational Agency', *Logic Journal of the IGPL*, 11(6), 597–613.

Gabbay, D.M., and Woods, J. (2003b), 'The Law of Evidence and Labelled Deductive Systems', *Phi-News*, 4, 5–46. http//phinews.ruc.dk/phinews4.pdf (Also Chapter 15 in Gabbay D.M., Canivez, P., Rahman, S., and Thiercelin, A. (Eds.), *Approaches to Legal Rationality, Logic, Epistemology, and the Unity of Science* 20, pp. 295–331, 1st ed., Springer, 2010, doi: 10.1007/978-90-481-9588-6_15).

Gaggl, S.A., and Woltran, S. (2010), CF2 Semantics Revisited. In eds. P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, *COMMA 2010*, vol. 216, IOS Press, Amsterdam pp. 243–254.

Haenni, R. (2009), 'Probabilistic Argumentation', *Journal of Applied Logic*, 7(2), 155–176.

Hajek, P., and Valdes, J. (1994), 'An Analysis of MYCIN-like Expert Systems', *Mathware and Soft Computing*, 1, 45–68.

Hajek, P., Havranek, T., and Jirousek, R. (1992) *Uncertain Information Processing in Expert Systems*, CRC Press. Taylor and Francis, Abingdon UK.

Hanh, D.D, Dung, P.M., and Thang, P.M. (2011), 'Inductive Defense for Modgil's Extended Argumentation Framework', *Journal of Logic and Computation*, 21(2), 307–349.

Hunter, A. (2001), 'Ramification Analysis with Structured News Reports using Temporal Argumentation', *Proc. of the Adventures in Argumentation Workshop (at 6th ECSQARU)*. http://www.cs.ucl.ac.uk/staff/a.hunter/papers/ra2.ps.

Jakobovits, H., and Vermeir, D. (1999), 'Robust Semantics for Argumentation Frameworks', *Journal of Logic and Computation*, 9, 215–161.

Keynes, J.M. (1973), *A Treatise on Probability*, Macmillan Press for the Royal Economic Society (1st ed., 1921; paperback edition, Cambridge University Press, 1988).

Leite, J., and Martins, J. (2011) 'Social Abstract Argumentation', in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence 2011*, ed. T. Walsh IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence – Volume 22, pp. 2287–2292 AAAI Press ©2011.

Levin, S.A. (ed.) (1978), *Studies in Mathematical Biology*, Part II, *Populations and Communities*, Mathematical Association of America. Washington DC, USA.

Lotka, A.J. (1925), *Elements of Physical Biology*, Baltimore: Williams & Wilkins Co.

Mann, N., & Hunter, A. (2008), *Argumentation Using Temporal Knowledge*, COMMA 2008, Toulouse, France: IOS Press, May 28–30, pp. 204–215.

Martinez, D., Garcia, A., and Simari, G. (2008), 'An Abstract Argumentation Framework with Varied-Strength Attacks', in *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08).*

Matt, P.-A., and Toni, F. (2008), 'A Game-Theoretic Measure of Argument Strength for Abstract Argumentation', in *LOGICS IN ARTIFICIAL INTELLIGENCE Lecture Notes in Computer Science*, JELIA 2008. (vol. 5293), pp. 285–297.

May, R.M.C. (1976), 'Simple Mathematical Models with Very Complicated Dynamics', *Nature*, 261, 459–475.

Metcalfe, G., Olivetti, N., and Gabbay, D. (2008), *Proof Theory for Fuzzy Logics (Monograph)*, Springer. Berlin Heidelberg.

Modgil, S. (2009), 'Reasoning about Preferences in Argumentation Frameworks', *Artifical Intelligence*, 173(9–10), 901–934.

Modgil, S., and Bench-Capon, T.J.M. (2008), 'Integrating Object and Metal-Level Value Based Argumentation', in *Computational Models of Argument*, *Proceedings of COMMA 2008*, pp. 240–251.

Modgil, S., and Bench-Capon, T.J.M. (2010), 'Metalevel Argumentation', *Journal of Logic and Computation*, doi: 10.1093/logcom/exq054, first published online: 28 September 2010.

Murray, J.D. (2001), *Mathematical Biology* (Vol. 1), Springer-Verlag. Berlin Heidelberg.

Prakken, H. (2010), 'An Abstract Framework for Argumentation with Structured Arguments', *Argument and Computation*, 1, 93–124.

Rotstein, N.D., Moguillansky, M.O., Javier Garcia, A., and Ricardo Simari, G. (2010), 'A Dynamic Argumentation Framework', in *Proceedings of the 2010 conference on Computational Models of Argument: COMMA 2010*. IOS Press Amsterdam, The Netherlands, The Netherlands ©2010, pp. 427–438

Shafer, G. (1976), *Mathematical Theory of Evidence*, Princeton University Press. Princeton, New Jersey, USA.

Turchin, P. (2003), *Complex Population Dynamics*, Princeton University Press. Princeton, New Jersey, USA.

Volterra, V. (1926), 'Fluctuations in the Abundance of a Species Considered Mathematically', *Nature*, 118, 558–560.

Woods, J. (2004), *The Death of Argument: Fallacies in Agent-Based Reasoning*, Applied Logic Series, Dordrecht and Boston: Kluwer.

Appendix 1. Further examples of numerical networks

This appendix shows more familiar examples reviewed as numerical networks.

A.1. *Modal networks*

We can read the nodes as possible worlds in a Kripke model and read the values as fuzzy truth values. ε is the fuzzy value of the accessibility of a to b (i.e. *a arrow b* means a is a possible world for b (i.e. bRa holds), while x is the fuzzy value of a being a possible world in the first place. So if $V_e(\varphi)$ gives a fuzzy value to the wff φ at world e, then $V_b(\Box\varphi) = \mathbf{f}(V_b(\varphi), \bar{V}_{a_i}(\varphi), \varepsilon_i)$, where a_i are all the nodes leading with an arrow into b.

It is worth giving a formal definition. See Barringer, Gabbay, and Woods (2008) for full details.

Definition A.1

(1) Let $\mathbb{L}$ be a propositional language with atoms $\{q_1, q_2, \ldots\}$, a modality $\Box$ and possibly other connectives $\mathbb{C}$. To fix our thoughts, say $\mathbb{C} = \{\twoheadrightarrow, \neg\}$, where $\twoheadrightarrow$ can be thought of as the Łukasiewicz many-valued implication (with truth values in $[0, 1]$ and $0 =$ true and truth table [Value$(A \twoheadrightarrow B) =$ Max$(0, \text{Value}(B) - \text{Value}(A))$]) and $\neg$ is a negation (with truth table [Value$(\neg A) = 1 -$ Value(A)]).

(2) A modal network model $\mathbf{m}$ is a family of models $\mathbf{m}_q = (A, T, V_q, \mathbf{f}), q$ an atom of $\mathbb{L}$, such that each $\mathbf{m}_q$ is a finitely branching attack network model in the sense of Definition 2.4. Thus, in $\mathbf{m}$ A, T and $\mathbf{f}$ are fixed and V_q varies with q. We assume that $\mathbf{f}, V_q$ give values in $[0, 1]$. We take $\mathbf{f}(y, x_i, \varepsilon_i) = \mathsf{Sup}_i(\varepsilon_i \twoheadrightarrow x_i) = \mathsf{Sup}_i\ \text{Max}(0, x_i - \varepsilon_i)$.

(3) For each $t \in T$ and each wff φ, we define the value $V^n_\varphi(t)$, (for $n = 0, 1, 2, \ldots$) as follows:

   (a) $V^0_q(t) = V_q(t)$, for atomic q, and $t \in T$.

   (b) $V^{n+1}_q(t) = \mathbf{f}(V^n_q(t), \bar{V}^n_q(a_i), \bar{V}^n_q(a_i \looparrowright t))$, where $a_1, \ldots, a_n$ are all the nodes such that $a_i \looparrowright t \in T$.

   (c) $V^n_{A \twoheadrightarrow B}(t) = \text{Max}(0, V^n_B(t) - V^n_A(t))$.

   (d) $V^n_{\neg A}(t) = 1 - V^n_A(t)$.

   (e) $V^n_{\Box A}(t) = V^{n+1}_A(t)$.

   The reader should carefully note that we have huge scope here for defining a multitude of different modalities by choosing the dependence of $V^n_{\Box A}(t)$ on the set $\{V^{m+n}_A(t), m = 0, 1, \ldots\}$. What we here define is a **K**-type modality. We can also define the hypermodality of Gabbay (2002) by letting:

$$V^n_{\Box A}(t) = \begin{cases} V^{n+1}_A(t), & \text{for } n \text{ odd} \\ \text{Max}(V^{n+1}_A(t), V^n_A(t)), & \text{for } n \text{ even} \end{cases}$$

(4) We say $\mathbf{m}$ is stable iff for any wff A and any $t \in T$ there exists an n such that for all $m \geq n$ we have $V^n_A(t) = V^n_A(t)$. For stable models, we can let $V^\infty_A(t) = \mathsf{Lim}_i\ V^n_A(t)$.

(5) We call a stable model $(A, T, V^\infty_A, \mathbf{f})$, a fuzzy modal model for $\mathbb{L}$.

Example A.2 (Ordinary modal logic)

(1) Let (S, R, h) be a traditional Kripke model for the language with $\{\to, \neg, \Box\}$, with S the set of possible worlds, R the accessibility relation and h the assignment to the atoms, (i.e. for each atomic $q, h(q) \subseteq S$). We assume that (S, R) is finitely branching, i.e. for each t the set $S_t = \{s | tRs\}$ is finite. Note that many modal logics are complete for a class of finitely branching models.

(2) Let $A = S, T = S \cup \{a \looparrowright b | bRa\}$.

(3) Let $V_q(a \looparrowright b) = 1$ for all atomic q and let $V_q(t) = 1$ iff $t \in h(q)$, for $t \in S$.

(4) Let $\mathbf{f}(V(t), \bar{V}(a_i), \bar{V}(a_i \looparrowright t)) = 1$, where $a_1, \ldots, a_n$ are all nodes such that tRa_i holds, iff $\bar{V}(a_i) = 1$ for all $1 \leq i \leq n$.

(5) We claim this model is stable. This can be proved by induction on the wff φ.

(6) Note that we can get a new variety of modal logics by changing $\mathbf{f}$ from point to point, or by making $V^n_{\Box A}(t)$ dependent on $\{V^{n+m}_A(t) \mid m = 0, 1, 2 \ldots\}$ in a variety of ways.

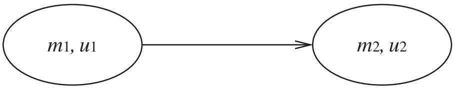


Figure A52. Collision of moving particles.

A.1.1. *Physics – special relativity*

This has to do with composition of masses and velocities in special relativity. Consider the situation in Figure A52

Particle 1 has rest mass m_1 and velocity u_1. Particle 2 has rest mass m_2 and velocity u_2. They are both moving in the same direction and they collide in an inelastic way, and form particle 3 with rest mass m_3 and velocity u_3. We view this as an attack of node (m_1, u_1) upon node (m_2, u_2) with relativistic transmission 1 (full transmission) to form a new value $(m_2', u_2') = (m_3, u_3)$. The new value is the following, see D'Inverno (1992, p. 51, exercise 4.7).

$$m_3^2 = m_1^2 + m_2^2 + 2m_2 m_2 \gamma_1 \gamma_2 \left(1 - \frac{u_1 u_2}{c^2}\right),$$
$$u_3 = \frac{m_1 \gamma_1 u_1 + m_2 \gamma_2 u_2}{m_1 \gamma_1 + m_2 \gamma_2},$$

with

$$\gamma_1 = \left(1 - \frac{u_1^2}{c^2}\right)^{-1/2}, \quad \gamma_2 = \left(1 - \frac{u_2^2}{c^2}\right)^{-1/2},$$

where, as usual, c is the speed of light.

Appendix 2. Interpretation of loops

There are various interpretations for the situation depicted in Figure 21 besides our argumentation network interpretation.

The ecology interpretation

The network can be interpreted as an ecology, where nodes represent species. Species a feeds on species b and species b feeds on species a. The functions $\mathbf{f}_1$ and $\mathbf{f}_2$ give the success rates. This is a predator–prey situation.

Let V_n be the population of some species at generation n. We assume population growth is a discrete process taking place in cycles. Such biological examples are provided by many temperate zone arthropod populations, with one short-lived adult generation each cycle. One possible recurrence equation is the following

$V_{n+1} = V_n(1 + r(1 - (V_n/K)))$, where r and K are constants.

K is the maximum size for the population and r is a factor measuring dependence on the density of the population. The reader should compare this equation with the equation $V_b = \mathbf{f}_1(V_a, x, \varepsilon)$ arising from Figure 21 (Levin 1978, p. 324).

This equation is called *the nonlinear logistic equation* which has the standard form

$$U_{n+1} = rU_n(1 - U_n), \quad r > 0.$$

This equation can exhibit chaotic behaviour depending on the value r (see Murray 2001).

A slightly different pair of equations has to do with parasitic life forms. Here we have, besides the population N_n, a parasitic population V_n. The recursive equations look like the following:

- $V_{n+1} = N_n - N_{n+1}/F$
- $N_{n+1} = FN_n f(N_n, V_n)$.

F is a factor indicating the proportion of those who escape the parasite. The difference between this equation for V_{n+1} and a direct recursion for V_{n+1} is that it is more complex. We get

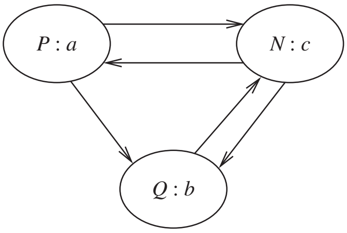


Figure A53. Predator prey network.

- $V_{n+1} = N_n(1 - f(N_n, V_n))$
- $N_{n+1} = FN_nf(N_n, V_n)$

See Levin (1978, p. 338).

Let us look at another example from biology. This is a model by Hassell (1978) of two parasitoids (P and Q) and one host (N) model. The equations are (see Anderson, Turner, and Taylor 1979, p. 295)

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

where N, P and Q denote the host and two parasitoid species in generations t and $t + 1$, 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 P_t or Q_t parasitoids, respectively. This model applies to two quite distinct types of interaction that are frequently found in real systems. It applies to cases where P acts first, to be followed by Q acting only on the survivors. Such is the case where a host population with discrete generations is parasitised at different developmental stages. In addition, it applies to cases where both P and Q act together on the same host stage, but the larvae of P always out-compete those of Q, should multi-parasitism occur.

The functions f_1 and f_2 are as follows:

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

where k_1 and k_2, a_1 and a_2 are constants.

To compare the biological model with the argumentation model, we put $a_1 = a_2 = 1$, $\lambda = 1$ and $k_1 = k_2 = -1$.

This gives

$$\begin{aligned} f_1(P_t) &= 1 - P_t \\ f_2(Q_t) &= 1 - Q_t \end{aligned}$$

and therefore

$$\begin{aligned} N_{t+1} &= N_t(1 - P_t)(1 - Q_t) \\ P_{t+1} &= P_t N_t \\ P_{t+1} &= Q_t N_t(1 - P_t) \end{aligned}$$

giving us the appropriate functions for attack and counterattack for the situation in Figure A53.

In Figure A53, a and b attack c. c counterattacks a and b and a attacks b. The transmission rates are 1. Since c is attacked by a and b, the new value for c is $N(1 - P)(1 - Q)$. Since a is counterattacked by c, the new value for a is PN (see Example 2.2 as to why the counterattack value is PN and not $(1 - P)N$). Since b is counterattacked by c and attacked by a the new value for b is $QN(1 - P)$.

To give Figure A53 some meaning, think of a, b, c as follows:

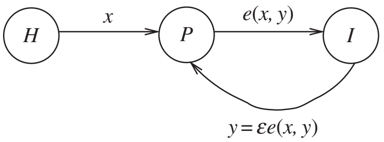


Figure A54. Feedback network.

$c =$ The US President has a strong case for re-election.
$a =$ A deteriorating situation in Iraq (US soldiers killed) (attacks his chances).
$b =$ Lack of success in combatting Al-Qaeda.

Clearly, if the value of c is low, less effort is required in attacking it from a and b. This explains the counterattack loop.

a attacks b by the argument that the situation in Iraq has diverted Al Qaeda away from attacking US territory proper.

To sum up, we have shown a connection with biological models. In view of this connection, we would like to refer to loops as *ecologies* (of arguments).

Feedback interpretation

We can consider the figures as a control-feedback situation. Say node b is a feedback for node a.

We consider a simple model of an electrical generator. We have three nodes H, P and I. H represents a hydraulic power source, moving a core in the power plant P producing electricity. The core rotates inside a magnetic field generated by a current coming from industry I. This produces a current which goes to I (Industry). I then sends part of the current it receives back to P and thus maintaining the magnetic field. Figure A54 describes the situation in its steady state.

To keep the story simple, let us assume that there is full transmission from H to P, from P to I and from I to P. Let x be the transmission of hydraulic energy coming from H and let y be the current coming from I. Let $e(x, y)$ be the resulting current transmitted to I and let ε be the proportion of it coming back from I to P. Thus e is monotonic increasing in x and y and we have in a steady state that

- $y = \varepsilon e(x, y)$
- $e(x, 0) = e(0, y) = 0$ (i.e. if no hydraulic power turns the core or no current is coming to maintain the electromagnetic field, then we have no output e.)
- ε is such that $(1 - \varepsilon)e$ is maximal (i.e. we feedback current in such a way that what is left at I is maximal, given a fixed x).