

# Introducing Equational Semantics for Argumentation Networks

Dov M. Gabbay

<sup>1</sup> King's College London

dov.gabbay@kcl.ac.uk

<sup>2</sup> Bar Ilan University Israel

<sup>3</sup> University of Luxembourg

**Abstract.** This paper provides equational semantics for Dung's argumentation networks. The network nodes get numerical values in  $[0,1]$ , and are supposed to satisfy certain equations. The solutions to these equations correspond to the "extensions" of the network.

This approach is very general and includes the Caminada labelling as a special case, as well as many other so-called network extensions, support systems, higher level attacks, Boolean networks, dependence on time, etc, etc.

The equational approach has its conceptual roots in the 19th century following the algebraic equational approach to logic by George Boole, Louis Couturat and Ernst Schroeder.

## 1 Introduction

This paper is a short version of [11], which expands (as promised) on our equational ideas introduced in pages 246–251 of [9]. The Equational approach has its conceptual roots in the 19th century following the algebraic equational approach to logic by George Boole [2], Louis Couturat [4] and Ernst Schroeder [13].

The equational algebraic approach was historically followed, in the first half of the 20th century by the Logical Truth (Tautologies) approach supported by giants such as G. Frege, D. Hilbert, B. Russell and L. Wittgenstein. In the second half of the twentieth Century the new current approach has emerged, which was to study logic through its consequence relations, as developed by A. Tarski, G. Gentzen, D. Scott and (for non-monotonic logic) D. Gabbay.

### 1.1 Aims of This Paper

We have several good reasons for writing [11], the full paper.

1. To provide a general computational framework for Dung's argumentation networks; a framework in which the logical aspects, computational aspects and the conceptual aspects involved in Dung's original proposal can be isolated, highlighted and analysed, and thus paving the way for orderly responsible generalisations.

The logical aspects involve the question of what is the logical content of an argumentation network and what inferences we can draw from it, see [8]. The computational aspects have to do with viewing the abstract argumentation networks

as directed graphs or as finite models with binary relations on them and various algorithms for extracting subsets of such graphs or models. See for example our paper [12] on annotation theories. The conceptual aspect is the reason behind the computation, involving concepts such as admissibility and a variety of extensions.

At present Dung's networks are generalised in chaotic and incompatible ways by many capable researchers. Unfortunately, we have no general meta-level approach which the community can use for guidance and comparison.

2. To generalise Dung's argumentation networks in a natural way and connect and compare it with other networks communities, such as neural nets, Bayesian nets, biological-ecological nets, logical labelled deductive nets and more.

These networks have a different conceptual base but they look like abstract argumentation networks, i.e. they are directed graphs. We manipulate the the graphs differently because they come from different applications. So the question to ask is whether we can we find common grounds (such as an equational approach to such graphs) which will bring the applications together at least on the formal mathematical side?

3. To introduce in a natural way various meta operations on networks such as distributed networks (modal logic), time dependence and fibring which exist in other types of networks and logics.
4. To connect with pure mathematics, numerical analysis and computational algebra.
5. To show the argumentation community the extent of our own contributions to this area and the rationale and priority of these contributions.<sup>1</sup>

Dung's argumentation networks (see [6]) have the form  $(S, R_A)$  where  $S$  is a set of arguments, which for the current purposes we assume to be finite, and  $R_A$  is a binary attack relation on  $S$ . We are interested in subsets  $E$  of  $S$  of arguments which are admissible, that is self defending and conflict free, namely:

1.  $E$  is conflict free, namely for no  $x, y$  in  $E$  do we have that  $xR_Ay$ .
2.  $E$  defends each of its elements: Whenever for some  $x$ , we have  $xR_Ay$  and  $y$  is in  $E$ , there is some  $z$  in  $E$  defending  $y$ , i.e. we have  $zR_Ax$ . ( $E$  is self-defending.)
3.  $E$  is complete if  $E$  contains all the elements it defends.

The smallest such  $E$  is called the *grounded extension*, a maximal  $E$  (there may be several different such maximal sets) is called a *preferred extension*, and if we are lucky, we may also have a *stable extension*  $E$ , namely one which attacks anything not on it.

See [5; 12] for surveys.

Such extensions are perceived as indicating coherent logical positions which can defend themselves against attacks.

We make use of the Caminada labelling functions  $\lambda : S \mapsto \{\text{in, out, undecided}\}$ .  $\lambda$  satisfies the following condition

---

<sup>1</sup> The argumentation area has undergone phenomenal expansion, and many researchers are not aware of the full extent of research done in the area, resulting in a poor record in crediting and quoting each others' work.

- (C1) If  $\lambda(a) = \text{in}$  and  $aR_A b$  then  $\lambda(b) = \text{out}$ .  
 (C2) If for all  $x$ ,  $(xR_A b \rightarrow \lambda(x) = \text{out})$  then  $\lambda(b) = \text{in}$ . (Note that this includes the case that  $b$  has no attackers.)  
 (C3) If  $\forall x(xR_A b \rightarrow \lambda(x) \neq 1)$  and  $[\exists y(yR_A b \text{ and } \lambda(y) = \text{undecided})]$  then  $\lambda(b) = \text{undecided}$ .

Every such  $\lambda$  gives a complete extension  $E_\lambda = \{x | \lambda(x) = \text{in}\}$ , and vice versa. See [5].

## 1.2 Equational Examples

This subsection is intended to motivate the formal equation section, Section 2. We give here several examples of the equational approach.

Let  $(S, R_A)$  be a Dung network. So  $R_A \subseteq S^2$  is the attack relation. We are looking for a function  $\mathbf{f} : S \mapsto [0, 1]$  assigning to each  $a \in S$  a value of  $0 \leq \mathbf{f}(a) \leq 1$  such that the following holds.

1.  $(S, R_A, \mathbf{f})$  satisfies the following equations for some family of functions  $\{\mathbf{h}_a\}$ ,  $a \in S$ :
  - (a) If  $a$  is not attacked (i.e.  $\neg \exists x(xR_A a)$ ) then  $\mathbf{f}(a) = 1$ .
  - (b) If  $x_1, \dots, x_n$  are all the attackers of  $a$  (i.e.  $\bigwedge_{i=1}^n x_i R_A a \wedge \forall y(yR_A a \rightarrow \bigvee_{i=1}^n y = x_i)$ ) then we have that  $\mathbf{f}(a) = \mathbf{h}_a(\mathbf{f}(x_1), \dots, \mathbf{f}(x_n))$ .

Let us take, for example the same  $\mathbf{h}_a = \mathbf{h}$  for all  $a$  and let

$$\mathbf{h}(\mathbf{f}(x_1), \dots, \mathbf{f}(x_n)) = \prod_{i=1}^n (1 - \mathbf{f}(x_i))$$

The above equation we shall call  $E_{q_{\text{inverse}}}$ . We shall define other possible equations later on.

Thus we get

$E_{q_{\text{inverse}}}$  for the function  $\mathbf{f}$ :

$$\mathbf{f}(a) = \prod_{i=1}^n (1 - \mathbf{f}(x_i)).$$

2. For any Caminada labelling  $\lambda$  of  $(S, R_A)$ , there exists an  $(S, R_A, \mathbf{f})$  such that

$$E_{\mathbf{f}} = \begin{cases} \text{If } \lambda(a) = \text{in} \text{ then } \mathbf{f}(a) = 1 \\ \text{If } \lambda(a) = \text{out} \text{ then } \mathbf{f}(a) = 0 \\ \text{If } \lambda(a) = \text{undecided} \text{ then don't care what } \mathbf{f}(a) \text{ is} \\ \text{provided it satisfies the equations.} \end{cases}$$

Condition (1) above reads  $\lambda(a) = \text{in}$  as  $\mathbf{f}(a) = 1$  and  $\lambda(a) = \text{out}$  as  $\mathbf{f}(a) = 0$ .

Therefore the equation

$$\mathbf{f}(a) = \prod_{i=1}^n (1 - \mathbf{f}(x_i))$$

ensures that:

- If one of  $x_i$  ( $x_i$  are the attackers of  $a$ ) is in then  $a$  is out.
- If all the attackers are out then  $a$  is in.

The question is what happens with the undecided cases. Here we have condition (2).

Any Dung extension can have a corresponding function  $\mathbf{f}$  which agrees with the “in” and “out”, though may be also more specific about the undecided.

So if the Dung extension says I don’t know, the function  $\mathbf{f}$  can say whatever it wants, provided it satisfies the equations.

Note that we can have a different function  $\mathbf{h}$ . Time to give a formal definition.

**Definition 1 (Possible equational systems).** *Let  $(S, R_A)$  be a networks and let  $a$  be a node and let  $x_1, \dots, x_n$  be all of its attackers.*

*We list below several possible equational systems, we write  $Eq(\mathbf{f})$  to mean the equational system  $Eq$  applied to  $\mathbf{f}$ :*

1.  $Eq_{\text{inverse}}(\mathbf{f})$

$$\mathbf{f}(a) = \prod_i (1 - \mathbf{f}(x_i))$$

2.  $Eq_{\text{geometrical}}(\mathbf{f})$

$$\mathbf{f}(a) = \left[ \prod_i (1 - \mathbf{f}(x_i)) \right] / \left[ \prod_i (1 - \mathbf{f}(x_i)) + \prod_i x_i \right].$$

*We call this equation  $Eq_{\text{geometrical}}$  because it is connected to the projective geometry Cross Ratio, see our 2005 paper [1].*

3.  $Eq_{\text{max}}(\mathbf{f})$

$$\mathbf{f}(a) = 1 - \max(\mathbf{f}(x_i)).$$

4.  $Eq_{\text{suspect}}(\mathbf{f})$

*We shall see the difference in the examples. In fact we shall see that this new function gives exactly the Caminada labelling.*

*Let us further introduce a fourth system of equations which we call  $Eq_{\text{suspect}}(\mathbf{f})$ :*

$$\mathbf{h}_a(\mathbf{f}(x_1), \dots, \mathbf{f}(x_n)) = \prod_i (1 - \mathbf{f}(x_i)), \text{ if } \neg aR_A a \text{ holds}$$

and

$$\mathbf{h}_a(\mathbf{f}(x_1), \dots, \mathbf{f}(x_n)) = \mathbf{f}(a) \prod_i (1 - \mathbf{f}(x_i)), \text{ if } aR_A a \text{ holds.}$$

*Example 1.* Let us do an example using all four options for equations, namely  $Eq_{\text{geometrical}}$ ,  $Eq_{\text{inverse}}$ ,  $Eq_{\text{max}}$  and  $Eq_{\text{suspect}}$ .

Consider Figure 1. We are looking for  $\mathbf{f}$  solving the equations. Let  $\mathbf{f}(a) = \alpha$ ,  $\mathbf{f}(b) = \beta$ ,  $\mathbf{f}(c) = \gamma$ .

I We use  $Eq_{\text{inverse}}$ :

The equations are

1.  $\alpha = (1 - \alpha)(1 - \gamma)$
2.  $\beta = (1 - \beta)(1 - \alpha)$
3.  $\gamma = 1 - \beta$

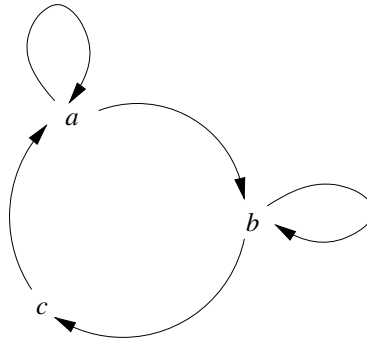


Fig. 1.

There are programs like Maple which can solve the equations of this sort and give all the solutions. We used one and got

$$\begin{aligned}\alpha &= 1 - \frac{\sqrt{2}}{2} \\ \beta &= \sqrt{2} - 1 \\ \gamma &= 2 - \sqrt{2}\end{aligned}$$

The interest in this case is that we are getting all kinds of values which shows that these equations are sensitive to the nature of the loops involved!

II. We use  $Eq_{\max}$ :

The equations are

1.  $\alpha = 1 - \max(\alpha, \gamma)$
2.  $\beta = 1 - \max(\beta, \alpha)$
3.  $\gamma = 1 - \beta$

The only solution in this case is  $\alpha = \beta = \gamma = \frac{1}{2}$ .

III. We use  $Eq_{\text{suspect}}$ :

The equations are

1.  $\alpha = \alpha(1 - \alpha) \cdot (1 - \gamma)$
2.  $\beta = \beta(1 - \beta)(1 - \alpha)$
3.  $\gamma = 1 - \beta$ .

The solution is  $\alpha = 0, \beta = 0, \gamma = 1$ .

IV. We use  $Eq_{\text{geometrical}}$ .

The equations are:

1.  $\alpha = \frac{(1-\alpha)(1-\gamma)}{(1-\alpha)(1-\gamma)+\alpha\gamma}$
2.  $\beta = \frac{(1-\alpha)(1-\beta)}{(1-\alpha)(1-\beta)+\alpha\beta}$
3.  $\gamma = 1 - \beta$ .

The only solution is  $\alpha = \beta = \gamma = \frac{1}{2}$ .

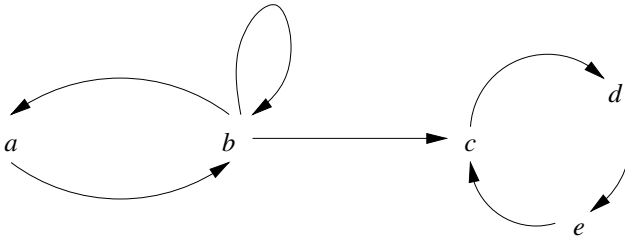
*Example 2 (Comparing  $Eq_{\max}$  and  $Eq_{\text{inverse}}$ ).* We shall show that these two equational systems may not yield the same extensions. the network is described in Figure 2.

Extensions according to  $Eq_{\max}$ .

Let us compute the equations according to  $Eq_{\max}$  and their possible solutions.

The equations are (we write “ $x$ ” instead of  $f(x)$ ):

1.  $a = 1 - b$
2.  $b = 1 - \max(a, b)$
3.  $c = 1 - \max(b, e)$
4.  $d = 1 - c$
5.  $e = 1 - d$ .



**Fig. 2.**

We get two extensions

1.  $\{a\}, (a = 1, b = 0, c = d = e = \frac{1}{2})$
2.  $\phi, (a = \frac{1}{2}, b = \frac{1}{2}, c = d = e = \frac{1}{2})$

Compare this result with Theorem 2 below.

We now deal with Figure 2 using  $Eq_{\text{inverse}}$ . The equations are:

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

We can have only one extension

$$\{a\}, (a = 1, b = 0, c = d = e = \frac{1}{2}).$$

## 2 Formal Theory of the Equational Approach to Argumentation Networks

In this section we formally develop our equational approach. Conceptually the nodes and the Equations attached to them is the network and the solutions to the equations are the complete extensions, as we have seen in the examples of Section 1.

### Definition 2 (Real equational networks)

1. An argumentation base is a pair  $(S, R_A)$  where  $S \neq \emptyset$  is a finite set and  $R_A \subseteq S^2$ .
2. A real equation function in  $k$  variables  $\{x_1, \dots, x_k\}$  over the real interval  $[0, 1]$  is a continuous function  $\mathbf{h} : [0, 1]^k \mapsto [0, 1]$  such that
  - (a)  $\mathbf{h}(0, \dots, 0) = 1$
  - (b)  $\mathbf{h}(x_1, \dots, 1, \dots, x_k) = 0$
 Sometimes we also have condition (c) below, as in ordinary Dung networks, but not always.
  - (c)  $\mathbf{h}(x_1, \dots, x_k) = \mathbf{h}(y_1, \dots, y_k)$  where  $\{y_j\} = \{x_i\}$  are permutations of each other.
3. An equational argumentation network over  $[0, 1]$  has the form  $(S, R_A, \mathbf{h}_a)$ ,  $a \in S$  where
  - (a)  $(S, R_A)$  is a base
  - (b) For each  $a \in S$ ,  $\mathbf{h}_a$  is a real equation function.
  - (c) If  $\neg \exists y(yR_A a)$  then  $\mathbf{h}_a \equiv 1$ .
4. An extension is a function  $\mathbf{f}$  from  $S$  into  $[0, 1]$  such that the following holds:
  - $\mathbf{f}(a) = 1$  if  $\neg \exists y(yR_A a)$
  - If  $\{x_1, \dots, x_k\}$  are all the elements in  $S$  such that  $x_i R_A a$ , then  $\mathbf{h}_a$  is a  $k$  variable function and  $\mathbf{f}(a) = \mathbf{h}_a(\mathbf{f}(x_1), \dots, \mathbf{f}(x_k))$ .

**Theorem 1 (Existence theorem).** Let  $(S, R_A, \mathbf{h}_a)$ ,  $a \in S$  be a network as in Definition 2. Then there exists an extension function  $\mathbf{f}$  satisfying (4) of Definition 2.

*Proof.* Let  $n$  be the number of elements of  $S$ . For each  $a \in S$  consider  $\mathbf{h}_a$  as a continuous function from  $[0, 1]^S \mapsto [0, 1]$ . Let  $\mathbf{h}$  be the continuous function from  $[0, 1]^S$  into  $[0, 1]^S$  defined component wise by  $\mathbf{h}(\alpha_1, \dots, \alpha_n) = (\mathbf{h}_{a_1}(\alpha_1, \dots, \alpha_n), \dots, \mathbf{h}_{a_n}(\alpha_1, \dots, \alpha_n))$ .

This is a continuous function on a compact cube of  $n$  dimensional space and has therefore, by Brouwer's fixed point theorem, a fixed point  $(x_1, \dots, x_n) = \mathbf{h}(x_1, \dots, x_n)$ .

Let  $\mathbf{f}$  be defined by  $\mathbf{f}(a_i) = x_i$ . Then we have that for each  $a \in S$

$$\mathbf{f}(a) = \mathbf{h}_a(\mathbf{f}(a_1), \dots, \mathbf{f}(a_k))$$

where  $a_i$  are all the points in  $S$  attacking  $a$ .

*Remark 1.* For Brouwer's fixed point theorem see Wikipedia.<sup>2</sup>

**Lemma 1.** Let  $(S, R_A)$  be a Dung argumentation network. Let  $\lambda : S \mapsto \{\text{in}, \text{out}, \text{undecided}\}$  be a legitimate Caminada labelling, yielding an extension  $E_\lambda$ . Consider the functions  $\mathbf{h}_a$ ,  $a \in S$  as follows:

<sup>2</sup> See [http://en.wikipedia.org/wiki/Brouwer\\_fixed\\_point\\_theorem](http://en.wikipedia.org/wiki/Brouwer_fixed_point_theorem) and Sobolev, V. I., "Brouwer theorem", in Hazewinkel, Michiel, "Encyclopaedia of Mathematics, Springer, 2001.

1.  $\mathbf{h}_a \equiv 0$  if  $\lambda(a) = \text{out}$
2.  $\mathbf{h}_a \equiv 1$  if  $\lambda(a) = \text{in}$ .
3.  $\mathbf{h}_a$  arbitrary real equation function, otherwise.

Then there exists, by Theorem 1 an extension function  $\mathbf{f}$  such that for all  $a \in S$

$$\mathbf{f}(a) = \mathbf{h}_a(\mathbf{f}(x_1), \dots, \mathbf{f}(x_k)),$$

where  $\{x_i\}$  are all the nodes attacking  $a$ .

Note that we have argued in these examples that we get a good refinement of the undecided allocations.

To get exactly the Caminada labelling, we use the next theorem, Theorem 2.

**Theorem 2 (Caminada labelling functions and  $Eq_{\max}$ ).** Consider the function

$$\mathbf{h}_{\max}(x_1, \dots, x_n) = 1 - \max(x_1, \dots, x_n).$$

This function is continuous in  $[0, 1]^n \mapsto [0, 1]$  and therefore falls under Definition 2.

1. Let  $(S, R_A, \mathbf{h}_{\max})$  be an equational network with  $\mathbf{h}_{\max}$  and let  $\mathbf{f}$  be an extension, as in item 4 of Definition 2. Define a labelling  $\lambda_{\mathbf{f}}$  dependent on  $\mathbf{f}$  as follows

$$\lambda_{\mathbf{f}}(a) = \begin{cases} \text{in} & \text{if } \mathbf{f}(a) = 1 \\ \text{out} & \text{if } \mathbf{f}(a) = 0 \\ \text{undecided} & \text{if } 0 < \mathbf{f}(a) < 1 \end{cases}$$

Then  $\lambda_{\mathbf{f}}$  is a proper Caminada extension of  $(S, R_A)$ .

2. Let  $\lambda$  be a Caminada extension for  $(S, R_A)$ . Let  $\mathbf{f}_{\lambda}$  be the real number function defined as follows

$$\mathbf{f}_{\lambda}(a) = \begin{cases} 1 & \text{if } \lambda(a) = \text{in} \\ 0 & \text{if } \lambda(a) = \text{out} \\ \frac{1}{2} & \text{if } \lambda(a) = \text{undecided}. \end{cases}$$

Then  $\mathbf{f}_{\lambda}$  is a proper equational extension for  $(S, R_A, \mathbf{h}_{\max})$ , i.e.  $\mathbf{f}_{\lambda}$  solves the equations  $\mathbf{f}_{\lambda}(a) = 1 - \max(\mathbf{f}_{\lambda}(x_1), \dots, \mathbf{f}_{\lambda}(x_n))$  where  $x_i$  are all the attackers of  $a$ .

*Proof.*

1. We show that  $\lambda_{\mathbf{f}}$  satisfies the Caminada conditions (C1)–(C3).

**Case C1** Assume  $x_1$  attacks  $a$  and  $\lambda_{\mathbf{f}}(x_1) = \text{in}$ . This means that  $\mathbf{f}(x_1) = 1$ . Let  $x_2, \dots, x_n$  be the other attackers of  $a$ . Then  $\mathbf{f}(a) = 1 - \max(\mathbf{f}(x_1), \dots, \mathbf{f}(x_n))$  and hence  $\mathbf{f}(a) = 0$  and hence  $\lambda_{\mathbf{f}}(a) = \text{out}$ .

**Case C2** Assume  $a$  has no attackers then  $\mathbf{f}(a) = 1$  and  $\lambda_{\mathbf{f}}(a) = \text{in}$ .

Otherwise let as before  $x_1, \dots, x_n$  be all the attackers of  $a$ , and assume  $\lambda_{\mathbf{f}}(x_i) = \text{out}$ , for all  $i$ . This means  $\mathbf{f}(x_i) = 0$  for all  $i$ . Hence  $\max(\mathbf{f}(x_i)) = 0$  and hence  $\mathbf{f}(a) = 1$  and hence  $\lambda_{\mathbf{f}}(a) = \text{in}$ .

**Case C3** Assume  $\lambda_{\mathbf{f}}(x_i) = \text{out}$  or undecided, with say  $\lambda_{\mathbf{f}}(x_1)$  at least is undecided. This means that  $\mathbf{f}(x_i) < 1$  for all  $i$  and for at least  $x_1$  we have  $\mathbf{f}(x_1) > 0$ . This means that  $0 < \max(\mathbf{f}(x_i)) < 1$ .

Hence  $0 < 1 - \max(\mathbf{f}(x_i)) < 1$ . Hence  $0 < \mathbf{f}(a) < 1$ . Hence  $\lambda_{\mathbf{f}}(a) = \text{undecided}$ .



2. Let  $\lambda$  be a proper Caminada extension. We show that  $\mathbf{f}_\lambda$  solves the equations with  $\mathbf{h}$ .

(a) If  $a$  has no attackers then  $\lambda(a) = \text{in}$  and  $\mathbf{f}_\lambda(a) = 1$ .

(b) Let  $x_1, \dots, x_n$  be all attackers of  $a$ .

i. If for some  $i$ ,  $x_i = \text{in}$  then  $\mathbf{f}_\lambda(x_i) = 1$ .

Also in this case  $\lambda(a) = 0$  and so  $\mathbf{f}_\lambda(a) = 0$ .

But  $\max(\mathbf{f}(x_i)) = 1$  and hence indeed  $\mathbf{f}_\lambda(a) = 1 - \max(\mathbf{f}(x_i))$ .

ii. If all  $\lambda(x_i) = \text{out}$  then  $\lambda(a) = \text{in}$ . So  $\mathbf{f}_\lambda(a) = 1$  and  $\mathbf{f}_\lambda(x_i) = 0$ . Thus  $\max(\mathbf{f}_\lambda(x_i)) = 0$ . So indeed  $\mathbf{f}_\lambda(a) = 1 - \max(\mathbf{f}_\lambda(x_i))$ .

(c) If all  $\lambda(x_i)$  are either out or undecided with at least  $\lambda(x_1) = \text{undecided}$  then  $\lambda(a) = \text{undecided}$  and so all  $\mathbf{f}_\lambda(x_i)$  are either 0 or  $\frac{1}{2}$  with at least  $\mathbf{f}_\lambda(x_1) = \frac{1}{2}$ , and  $\mathbf{f}_\lambda(a) = \frac{1}{2}$ .

Hence  $\max(\mathbf{f}(x_i)) = \frac{1}{2}$  and indeed  $\mathbf{f}_\lambda(a) = 1 - \max(\mathbf{f}_\lambda(x_i))$ .

*Remark 2 (Caminada labelling and  $Eq_{\text{inverse}}$ ).* Theorem 2 does not hold for  $Eq_{\text{inverse}}$ . This follows from Example 2.

**Summary 3 (Advantages of the Equational Approach).** *Time to list the advantages of our approach, see also Remark 3 below:*

*First let us highlight the fact that given a traditional argumentation network with attacks only, we use equations as a conceptual framework. We no longer talk about concepts like defense, acceptability, admissible extensions, etc. etc., but talk instead about solutions to the equations.*

*Therefore conceptually we have*

- *an extension is a solution to the equations and different extensions (grounded, preferred, stable, semi-stable, etc.) are characterised by further equations on these solutions functions using say Lagrange Multipliers see Section 6 below.*

*Within this framework we note the following:*

1. *To find all possible extensions we solve equations. We feed the equations into existing well known mathematical programs such as MAPLE or MATLAB or NSolve and get the solutions.*

*There are many papers which calculated computational complexity of finding various extensions, when we reduce the problem to that of solving equations in MAPLE or MATLAB or NSolve, complexity is not reduced, it can only increase.*

*What do we gain then?*

- *A new uniform framework, not only for argumentation networks, but also for other types, Ecological, flow, etc., etc.*
- *Possibility of finding different heuristics for equations which will work faster for most cases, giving an advantage over non-equational algorithms*
- *Ordinary people such as lawyers etc., to the extent that they use argumentation at all and are not averse to formal logic, they may find that it is psychologically easier to plug the problem into the computer, go and make a cup of tea and then check the results.*

Furthermore, if we insist on certain arguments being in or out, we can experimentally feed this into the equations and test the effect on other arguments. *MATLAB* itself does not generate all the solutions automatically but requires initial input, which is an advantage if we have a special set of arguments in mind.

For example the question of whether a set of arguments belongs to some extension (being credulous) of a certain type or whether the set belongs to all extensions of a certain type (being sceptical) can very naturally be handled in the equational framework.

To generate all extensions we need to keep plugging initial conditions into *MATLAB*, i.e., plug in all possible candidates for extensions (this is exponential in the number of nodes but we show in the full paper [11] that any Boolean set of functions can be embedded in argumentation networks, and so the complexity is exponential anyway).

Another possibility is to use *NSolve* which does generate solutions, see <http://reference.wolfram.com/mathematica/ref/NSolve.html>.

Another disadvantage of this is that we might get approximate solutions. So if we get  $x = 0.999$  we ask is this for real or is the solution supposed to be  $x = 1$ ?

On the other hand an advantage of using such programs is that it makes it easy to incorporate argumentations feature into other larger AI programs, as almost anything allows for solving equations.

2. We have a framework for introducing support discussed in the full paper [11].

### 3 Numerical Calculations

This section deals with numerical and computational aspects of our equational models.

We begin with options for calculating extensions in ordinary Dung networks and their comparison with Caminada labelling. Our embarkation point is a table from Caminada and Gabbay [5].

See Table 1.

We now write equations whose solutions give the correct extensions. We assume a set of equations  $Eq$  which is sound for Dung semantics, such as offered in Definition 1. Our network is  $(S, R_A)$ .

**Table 1.** Argument labellings and Dung-style semantics

restriction complete labellings	Dung-style semantics	linked by def. and th. of paper [5]
no restrictions	complete semantics	Def. 5 and Th. 1
empty undec	stable semantics	Def. 8 and Th. 5
maximal in	preferred semantics	Def. 10 and Th. 7
maximal out	preferred semantics	Def. 10 and Th. 7
maximal undec	grounded semantics	Def. 9 and Th. 6
minimal in	grounded semantics	Def. 9 and Th. 6
minimal out	grounded semantics	Def. 9 and Th. 6
minimal undec	semi-stable semantics	Def. 11 and Th. 8

**Case complete extensions**

Solve the equations. Any solution  $\mathbf{f}$  is an extension.

**Case stable extensions**

Add a new variable  $y$  such that  $y \notin S$ , and write the additional equation

$$\mathbf{f}(y) = \mathbf{h}_y = \sum_{x \in S} \mathbf{f}(x)(1 - \mathbf{f}(x)).$$

If the solution  $\mathbf{f}$  to the new expanded set of equations is a stable extension, then  $\mathbf{f}(x)(1 - \mathbf{f}(x))$  is 0 for all  $x \in S$  and hence  $\mathbf{f}(y) = 0$ . Conversely, if  $\mathbf{f}(y) = 0$  then  $\mathbf{f}$  is stable. Thus to check for stable extensions, we check  $\mathbf{f}(y)$ .

**Case of semi-stable extensions**

This case minimises the undecided. We do the following.

Consider the quantity

$$\mu = \sum_{\substack{a \in S \\ x_1, \dots, x_n \in S \\ \text{are all} \\ \text{attackers of } a}} [a - \mathbf{h}_a(x_1, \dots, x_n)]^2.$$

In  $\mu$  we regard all elements of  $S$  as variables. The equation  $\mu = 0$  has a solution. We regard  $\mu = 0$  as a constraint and minimise the expression

$$v = \sum_{x \in S} x(1 - x)$$

subject to the constraint  $\mu = 0$ .

This can be done using the method of Lagrange multipliers (see Wikipedia).

**Case of grounded extensions**

This is like the semi-stable case except that we minimise the expression  $1 - v$ .

**Case of preferred extensions**

This case is dealt with in the full paper. It is a bit involved and is of exponential complexity.

## 4 Equational Approach to Logic

We explain the general idea via some examples, and this would give the reader a better perspective on our equational approach to argumentation networks.

*Example 3 (Disjunctive inference).* Consider a simple inference:

1.  $(p \vee q)$
2.  $p \rightarrow r$
3.  $q \rightarrow r$

From (1)–(3) we want to infer

4.  $r$

We proceed as follows, assuming our logic satisfies the Deduction theorem:

$E$  and  $x$  proves  $y$  iff  $E$  proves  $x \rightarrow y$ .

5. 1. Assume  $p$
2. Get  $r$  from (5.1) and (2) using modus ponens.
6. 1. Assume  $q$
2. Get  $r$  from 6.1 and (3), using modus ponens
7. Get  $r$  from (1), (5-1–5.2) and (6.1–6.2) and the rule for disjunction elimination.

We now compare this with an equational approach.

Note that the above proof theoretic inference is valid in many logics, such as classical logic, intuitionistic logic and Łukasiewicz infinite valued logic.

When we write equations for the above inference, we have to choose in which logic we are operating. There will be different equations for different logics.

**Definition 3 (Boolean negation disjunction network).** *Let  $(S, R_-, R_+)$  be a network with two binary relations and the following properties:*

*Let  $R = R_- \cup R_+$ . Then the following holds*

1.  $xR_-y \wedge xR_-y' \rightarrow y = y'$ .
2.  $\neg \exists y_1 y_2 (xR_-y_1 \wedge xR_+y_2)$
3.  $xR_+y \rightarrow \exists ! z (z \neq y \wedge xR_+z)$ .
4.  $\neg xRx$ .

*We associate the following functions with  $R_+$  and  $R_-$ .*

1. *If  $xR_-y$  then  $x = 1 - y$ .*
2. *If  $xR_+y_1 \wedge xR_+y_2$  then  $x = [1 - (1 - y_1)(1 - y_2)]$ .*

*Example 4 (Equational approach to disjunctive inference in classical logic).* Consider Figure 3. This is a construction tree for the wffs involved in Example 3 from the point of view of classical logic. In classical logic  $R_+$  indicates disjunction.

Let us apply our equational definition of Definition 3 to Figure 3. We get in terms of  $p, q, r$  the following equations

1.  $u = [1 - (1 - p)(1 - q)]$
2.  $z = [1 - p(1 - r)]$
3.  $y = [1 - q(1 - r)]$

The disjunctive inference problem of Example 3 becomes the following equational problem

- Given that  $u = z = y = 1$ , solve for  $r$ .  $r = ?$

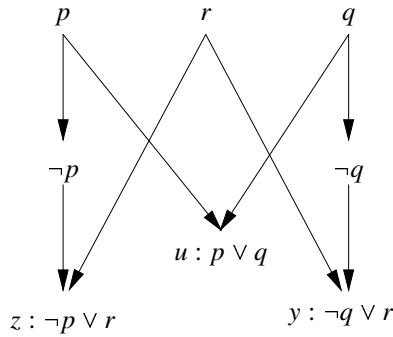


Fig. 3. F73

Let us see how to do it. We get

$$(*1) \quad (1 - p)(1 - q) = 0$$

$$(*2) \quad p(1 - r) = 0$$

$$(*3) \quad q(1 - r) = 0$$

The way the proof procedure of Example 3 proceeds is to do case analysis. From (\*1) either  $p = 1$  or  $q = 1$  and in each case from (\*2) (resp. \*3) we get  $r = 1$ .

This is not equational solving but reasoning about the equations to prove that  $r = 1$ .

We want to be more direct. Let us expand the equations.

$$(*1) \quad 1 - p - q + pq = 0$$

$$(*2) \quad p - pr = 0$$

$$(*3) \quad q - qr = 0$$

Add up all three equations and get

$$(*4) \quad 1 + pq - pr - qr = 0$$

or

$$(*5) \quad 1 + pq = (p + q)r.$$

Let us now add (\*2) and (\*3), we get

$$(*6) \quad p + q - pr - qr = 0$$

or

$$(*7) \quad (p + q) = (p + q)r.$$

We need to show that  $p + q$  is not 0 so that we can divide by it.

From (\*7) and (\*5) we get

$$(*8) \quad 1 + pq = p + q$$

We can also deduce from (\*8) that  $p + q \neq 0$  and so we divide by  $p + q$ .

So from (\*7) by dividing by  $p + q$  we get

(\*9)  $r = 1$ .

*Example 5 (Equational approach to disjunctive inference in Łukasiewicz many valued logic).* Łukasiewicz logic is formulated using  $\rightarrow$  and  $\neg$ , with the following truth table:

1. atoms get values in  $[0, 1]$
2.  $x \rightarrow y = \min(1, 1 - x + y)$
3.  $\neg x = 1 - x$
4. A wff is a tautology iff its truth value is always 1
5. Define  $x \oplus y = \neg x \rightarrow y$  and so we have

$$\neg x \rightarrow y = \min(1, x + y)$$

We can define therefore

$$x_1 \oplus \dots \oplus x_{n+1}$$

to be

$$\neg x_1 \rightarrow (\neg x_2 \rightarrow \dots \rightarrow (\neg x_n \rightarrow x_{n+1}) \dots)$$

and its table is

$$\min(1, x_1 + \dots + x_{n+1}).$$

Consider now the network of Definition 3. We use new functions for the case of  $xR_+y_1$  and  $xR_+y_2$ , we let

$$x = \min(y_1 + y_2).$$

We note that in Łukasiewicz logic the disjunction  $x \vee y$  has the table

$$x \vee y = \max(x, y)$$

and can be defined as

$$(x \rightarrow y) \rightarrow y.$$

We can define conjunction  $x \wedge y$  but

$$x \wedge y = \neg(\neg x \vee \neg y)$$

we have:

$$x \wedge y = \min(x, y).$$

6. The consequence relation for Łukasiewicz logic can be defined in several ways. We use the options which allows for the Deduction theorem, because the disjunctive proof in Example 3 uses it.

–  $A_1, \dots, A_n \vdash B$  iff  $(A_1 \oplus \dots \oplus A_n) \rightarrow B$  is a tautology.

The above means that  $\sum_i \text{value}(A_i) \leq \text{value}(B)$ .

Consider now the network of Figure 4.

Note that  $R_+$  in the figure indicates the connective  $x \oplus y = \neg x \rightarrow y$ , therefore we have

$$x \rightarrow y = (\neg x) \oplus y.$$

and disjunction  $x \vee y$  is defined as  $(x \rightarrow y) \rightarrow y$ , therefore

$$x \vee y = [\neg(\neg x) \oplus y] \oplus y.$$

We ask the following question:

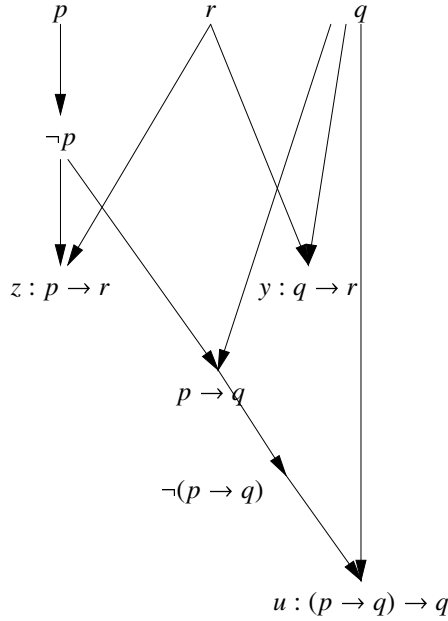


Fig. 4. F73a

– Given that  $\min(1, u + z + y) \leq r$ , what values can  $r$  have?  
 This means that given

$$\min(1, \max(p, q) + \min(1, 1 - p + r) + \min(1, 1 - q + r)) \leq r$$

what can  $r$  be?

Assume without loss of generality that  $p \geq q$ .

We ask can  $r$  be less than 1?

Assume  $r < 1$  and get a contradiction.

**Case 1**

$r < q \leq 1$ .

In this case we get

$$(p + 1 - p + r + 1 - q + r) \leq r$$

$$1 + 2r - q \leq r$$

$$2 + 2r \leq r + q$$

Since  $r < q$  we get

$$2 + r + q \leq r + q$$

$$2 \leq 1$$

not possible.

**Case 2**

$1 > r \geq q$

We get

$$\begin{aligned} p + 1 - p + r + 1 &\leq r \\ 1 + 1 &\leq 1 \end{aligned}$$

not possible.

Therefore  $r = 1$  and so the left hand side is

$$[\min(1, p + 1 + 1 - 1)] \leq [r = 1]$$

*Remark 3 (Equational reasoning).* Let us put on our meta-level hat and analyse what is happening here.

1. The logical database  $\Delta = \{p \vee q, p \rightarrow r, q \rightarrow r\}$  became a set of nodes  $E_\Delta$  in the network of Figure 3..
2. The inference problem
  - $\Delta \vdash? r$
 becomes the following equational question
  - If the points in  $E_\Delta$  all have value 1 does  $r$  have to get value 1 too?
 or more generally, let  $\mathbf{e}$  be a function assigning values to points in  $E$  and let  $y \notin E$ . We can ask
  - If  $E$  gets the values indicated by the function  $\mathbf{e}$ , what values are forced on  $y$ ?
3. The equational question in (2) is meaningful for any network. Take for example an argumentation network and take any set of nodes  $E_0$  and  $y \notin E_0$ . We can ask
  - Let  $E \supseteq E_0$  be any extension of a certain type (say  $E$  a stable extension) are we forced to have  $y \in E$ ?

In which case we can write

- $E \Vdash y_{\text{stable}}$
4. What is the analogous feature in the case of logic to the notion of extension in argumentation networks?
 

We know that any set of nodes corresponds to a database. So the algorithms generating extensions correspond to a way of generating databases.
  5. The notion of “consistency” in logic corresponds to “having a solution” in an equational network.

Let  $\mathbf{e}$  be a function associating values to the points in  $E$ . Then  $(\mathbf{e}, E)$  is equationally consistent, iff there exists a solution  $\mathbf{f}$  to the equations such that  $\mathbf{f} \upharpoonright E = \mathbf{e}$ .

## 5 Conclusion

We have shown the reader some of what the equational approach can do. The full paper (which may as well become a book) contains a lot more material, including:

1. Equations for higher level attacks (attacks from argument nodes to attack arrows of [10])
2. Equations for logic programs
3. Connections with fuzzy logic and fuzzy argumentation
4. Translation (critical faithful embedding) of Boolean networks (also known as abstract dialectical frameworks in [3]) into Dung networks
5. Time dependent networks and equations, including attacks arising from argument decay



6. Approximate admissible extensions (where we accept arguments whose value is almost 1) and their relation to weighted argument systems of [7]
7. Analysis of support
8. Equational characterisation of loops
9. General meta-level considerations
10. Comparison with related literature.

## Acknowledgements

I am grateful to Martin Caminada, Nachum Dershowitz, Phan Minh Dung, David Makinson, Alex Rabinowich, and Serena Villata for helpful discussions.

## References

1. Barringer, H., Gabbay, D.M., Woods, J.: Temporal dynamics of support and attack networks. In: Hutter, D., Stephan, W. (eds.) *Mechanizing Mathematical Reasoning*. LNCS (LNAI), vol. 2605, pp. 59–98. Springer, Heidelberg (2005)
2. Boole, G.: *The Mathematical Analysis of Logic*, Cambridge and London (1847)
3. Brewka, G., Woltran, S.: Abstract dialectical frameworks. In: *Proc. KR 2010*, pp. 102–111. AAAI Press, Menlo Park (2010)
4. Couturat, L.: *The Algebra of Logic*. Open Court (1914)
5. Caminada, M., Gabbay, D.M.: A logical account of formal argumentation. *Studia Logica* 93(2-3), 109–145 (2009)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and  $n$ -person games. *Artificial Intelligence* 77, 321–357 (1995)
7. Dunne, P., Hunter, A., McBurney, P., Wooldridge, M.: Weighted argument systems. *Artificial Intelligence* 175, 457–486 (2011)
8. Gabbay, D.M.: Provability foundations for argumentation networks. *Studia Logica* 93(2-3), 181–198 (2009)
9. Gabbay, D.M.: Fibring argumentation frames. *Studia Logica* 93(2-3), 231–295 (2009)
10. Gabbay, D.M.: Semantics for higher level attacks in extended argumentation frames. Part 1: Overview. *Studia Logica* 93, 355–379 (2009)
11. Gabbay, D.M.: Equational approach to argumentation networks, 90pp (February 2011)
12. Gabbay, D.M., Szałas, A.: Annotation theories over finite graphs. *Studia Logica* 93(2-3), 147–180 (2009)
13. Schröder, E.: *Vorlesungen über die Algebra die Logik*, vol. 3, pp. 1890–1904. B. G. Tuebner, Leipzig; Reprints, Chelsea (1966); Thoemmes Press (2000)