

Abstract. This paper is part of a research program centered around argumentation networks and offering several research directions for argumentation networks, with a view of using such networks for integrating logics and network reasoning.

In Section 1 we introduce our program manifesto. In Section 2 we motivate and show how to substitute one argumentation network as a node in another argumentation network.

Substitution is a purely logical operation and doing it for networks, besides developing their theory further, also helps us see how to bring logic and networks closer together.

Section 3 develops the formal properties of the new kind of network and Section 4 offers general discussion and comparison with the literature.

Keywords: fibring argumentation frames, joint attacks, disjunctive attacks, critical sets, flow argumentation networks.

1. Overview

1.1. Logic and networks — a manifesto

In the past half century various formal tools have been proposed for the study of human behaviour in daily life. Such tools were developed in computer science, communication, artificial intelligence, language study, law, analytic philosophy, psychology and cognition, among others. Main among these tools are the formal logical systems (classical logic, non-monotonic logics, modal and temporal logics, etc., etc.) and various network models such as argumentation networks, neural networks, Bayesian networks, inheritance networks, and more. There is no unifying view for all these tools, and in fact they are developed by completely different international communities with very little common ground and communication and yet (see below) all of these features of human behaviour (logics and networks) do reside coherently in the individual human mind and enable him to function intelligently in his day-to-day activity.

There is some realisation among a few of these diverse communities that communication between them needs to take place and unifying principles are indeed sought. Unfortunately not much is known and certainly no coherent

and successful unifying view exists. The mission of this manifesto to provide such a view.

To explain what we have in mind, we start with a simple example.

EXAMPLE 1.1 (The Messy Room). Mother goes into her teenage daughter's bedroom. Her instant impression is that it is a big mess. There is stuff scattered everywhere.

Mother's impression is that it is not characteristic of the girl to be like this.

What has happened?

Conjecture: The girl has boyfriend problems.

Further Analysis: Mother noticed a collapsed shelf. Did the girl smash it? Upon further observation, mother notices that the pattern of chaos shows that a shelf has collapsed because of excessive weight and scattered everything around, giving the impression of a big mess. But, actually, it is not a mess, it does make some (gravitational) sense.

There are several modes of reasoning:

1. Neural nets type of reasoning.

She recognises the mess instantly, like we recognise a face.

2. Nonmonotonic deduction.

Mother reasons from context and her knowledge of her daughter that the girl is not disorganised like this. She asks 'what happened?'.

3. Abduction/conjecture.

She offers a reasonable explanation that the girl has boyfriend problems. This is common to that age.

4. She then applies a database AI deduction and recognises that the mess is due to gravity. This deduction is no longer a neural net impression. It is a careful calculation.

- 4*. Item (4) could have been a neural net impression.

For example, a man who sees many shelf collapsing mess cases may recognise the pattern like it were a face (in which case it would be a neural net-like recognition).

- 2*. Item (2) could have been a Bayesian network.

Clearly all of these reasoning tools are working together in the mother's mind. Can we give a unified model? What does it look like in principle?

Furthermore, suppose both mother and father have seen the room. Father may reach different conclusions about the girl and demand some action.

A dialogue, argumentation and negotiation between the parents will follow with a view to reaching a merged knowledge base and an agreed course of action.

The value of a unified model goes beyond just a unifying formal theory. Even if we take the view that each of these components model a different aspect of the human (constructed as a model for the purpose of installing on a computer or a robot) a unified theory can help extend their range of applicability and help integrate them better. But we hope for more. We hope that such a model built up carefully might give us a better insight on how people actually reason. Something of great interest to the philosopher, psychologist, linguist and cognitive scientist. A unified theory would be a better, sharper tool in their hands.

First let us list what systems (and communities studying them) are involved.

We have:

A. Networks:

neural nets, argumentation nets, Bayesian nets, fuzzy nets, biological predator-prey networks, transportation networks, flow networks, inheritance nets, mathematical graphs, Kripke models, Description logics, Electrical networks, legal jurisdiction nets, social networks, input-output nets, and more.

B. Logics:

classical logic, modal and temporal logics, nonmonotonic logics, logic programming, Labelled deductive systems, and more.

C. Mechanisms:

abduction, belief revision and merging, consistency/paraconsistency, meta-level *vs.* object level, and more.

D. Metalevel principles:

fibring and combining systems, communication between systems, and more.

A quick analysis of this task immediately shows how huge the problem is. There is a lot of work to be done just in providing a unifying view between networks, let alone making a connection with logics. The rich variety of networks and the different research communities supporting them have different underlying assumptions, different ranges of applicability¹ and different kinds of mathematics involved. How do we bring them together?

¹It is regrettable that the communities involved have not studied the range of applicability of their tools. To compare with, e.g. pain killers offered by the pharmaceutical community, it is always made clear in what circumstances one can or cannot use them.

A simple working procedure seems successful. We first identify characteristic movements in each kind of network and then see whether we can generalise the other networks with similar features. Iterating this process will hopefully lead us to a general notion of network which can specialise to the various existing networks. Once we have that, we can try and see how this general network notion can unify with ordinary logics and other mechanisms.

This procedure has the advantage of extending and generalising each network we work with in a meaningful way. Thus each research area will benefit incrementally. Our understanding of characteristic movement existing in one source network (e.g. loops, feedback, aggregation) will be enhanced by developing its counterpart manifestations in other target networks, with the added benefit of meaningful generalisations in the target networks.

The task is not only a scientific problem but also a social problem.

The diverse communities of research in particular networks (e.g. Bayesian community, neural community, argumentation community, logic, etc.) are all immersed in their own areas and are not likely to respond to unifying theories. Our scientific strategy must be such that it is compatible with the social situation and is likely to generate enthusiasm and response from a significant group of researchers. This is why we proceed from the ground up as described above. Furthermore, it is fortunate that one type of network, the argumentation network is open to further generalisation and research in a natural and large scale way. Features existing in other networks can be naturally (though not easily) brought into argumentation networks and these features do have a natural meaning there. Furthermore the argumentation community itself is open-minded and is in fact desiring to expand their field. Many of their individual members have background in logic and will therefore easily see and respond to a general unifying theory with logic. So our strategy is as follows.

1. Assume the current state of argumentation networks is S
2. Study other networks and identify features F_1, F_2, \dots existent in such networks but not in S .
3. Extend the theory of S to new S_F by incorporating and generalising all of these features in a meaningful way. This is not simple. It is real research which will be meaningful to the argumentation community. It will also generate further generalizations to be exported to other networks.
4. Emerge with a generalised theory of network, say $S_{\text{new}} \supset S_F$.

- 5. Specialise S_{new} into other networks (e.g. network T) to show that S_{new} is general enough to unify big chunks of other networks. Get, for example, T_{new} . Chances are that T_{new} will naturally and intuitively suggest new features to be exported back to S_{new} to form S_{new}^1 .
- 6. Generalise further to a unified theory with logic, call it S_L .
- 7. Iterate the entire process to obtain $S_{\text{new}}^2, T_{\text{new}}^2, S_L^2, \dots$ etc.

The following table lists features from other networks which can be imported into argumentation networks.

Network	Properties
Neural	Feedback loops. Numerical weights on arcs. Real number values, function approximation and learning emphasis.
Argumentation	Nodes attacking nodes. Also sometimes supporting nodes. Loops is an issue with currently no consensus. Logic programming way of thinking. Nodes have logical content.
Bayesian	Loops forbidden. Probabilistic approach.
Fuzzy	Real number values and aggregation.
Biological	Emphasis on loops and cycles through the loops.
Transportation	Emphasis on paths and costs through paths.
Flow	Emphasis on flows and counterflows through paths and nodes. Nodes are sources and Sinks.
Inheritance	Emphasis on persistence and exceptions.
Graphs	Mathematical theory. Counting of nodes, topology on connectivity. Pure mathematics point of view.
Kripke models	Propagation of values for evaluating formulas.
Description logic nets	Similar to Kripke models but with different emphasis. A fragment of predicate logic.

Network	Properties
Electrical networks	Feedback loops. Equational approach. Network generates equations to be solved.
Legal jurisdiction network	Movability of data across jurisdiction.
Social networks	Information propagation.
Input-output networks	Nodes are logic processors.

To give the reader an idea of what S_F might look like, we present a diagram, Figure 1 of a generalised argumentation network and discuss its features.

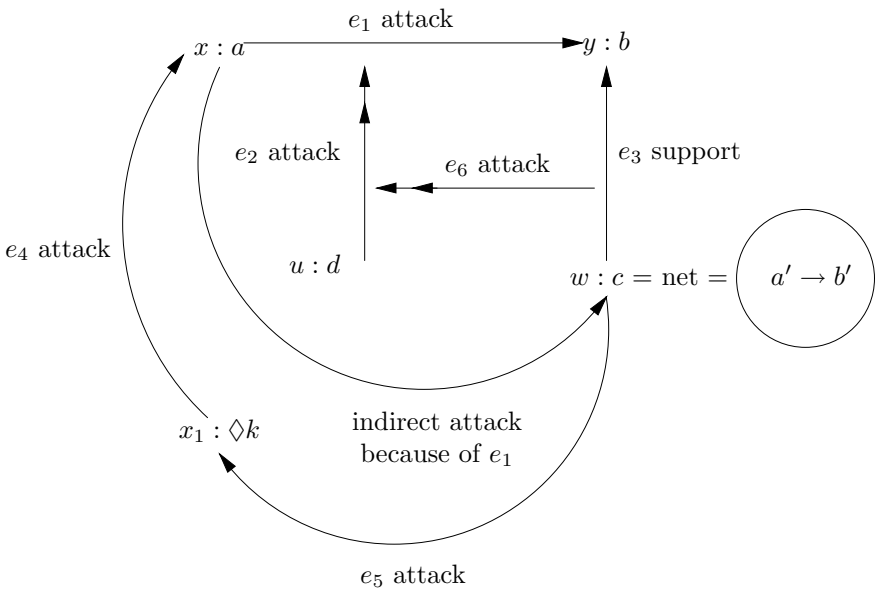


Figure 1.

- This is a complex diagram with the following generalisation:
1. It has both attack and support. Each argument has strength and there is a rate of transmission. For example, node c (strength w) supports b and the transmission rate is e_3 .
 2. There is feedback from nodes to arcs $u : d$ attacks the attack arc from $x : a$ to $y : b$.

3. Arcs can attack or support other nodes or arcs. E.g. the arc e_3 attacks the arc e_2 with strength e_6 .
4. There are indirect attacks, for example, of a on c because e supports b and a attacks b .
5. There are loops, because c attacks $x_1 : \Diamond k$ which attacks $x : a$. If c is weakened then the attack of $x_1 : \Diamond k$ may succeed and hence $x : a$ fails. So by attacking $y : b$, the node $x : a$ opens itself to attack from $x_1 : \Diamond k$ because it weakens $w : c$.
6. The entire system can be temporal if the strength and transmissions are time dependent and the language of the arguments is temporal. The node $x_1 : \Diamond k$ is temporal because x_1 depends on time and $\Diamond k$ says there is the (future) possibility of argument k .
7. The net contains a subnet. c is the subnet $a' \rightarrow b'$. So we can also fibre nets within nets. This is also a problem of communication between networks.
8. We need algorithms to propagate values within the net to get the emerging winning arguments. We need theory of aggregating values and handling loops. We need to combine with temporal logic and change and we need to know how to substitute one net inside the other.

Connection with logic

We will present logic data as networks. So the basic semantical notions will be in the diagram: In Logic

$$\text{formula, theory} \Vdash^{\text{evaluated}} \text{semantic model}$$

In our generalised S_L

$$\text{master network} \Vdash^{\text{evaluated}} \text{slave network}$$

This will include the logic case as a special case.

This is integrating with logic through the semantics. Proof theory can also be developed as rules of syntactical manipulation of networks. The framework for doing so is that of Labelled Deductive Systems [9].

To give the reader an idea of how we perceive logic as a network, consider the formula $p \wedge q \rightarrow r$. Its construction tree is Figure 2.

If we give values $p = 1, q = 1, r = 0$ to the atoms, then we can propagate them down the tree. Our conclusion from this that a formula A is a network. Now examine how we evaluate $\Diamond q$ at a 3-node Kripke model, Figure 3.

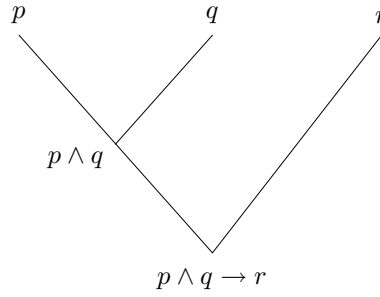


Figure 2.

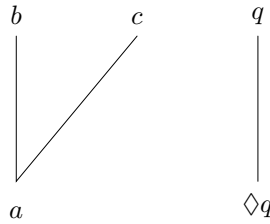


Figure 3.

First we give joint values of 0 or 1 to pairs (x, q) (meaning $x \models q$). Then we jointly and inductively propagate the values down both networks.

So if α is a node in one network and E is a node in another network we define the value $\alpha \models E$ by recursive induction on the known values of the rest of the pairs $\alpha' \models E'$. The ‘formulas’ correspond to one network and the ‘semantics’ is the other network. This way we can define the notion of one network being true in another. $M \models N_1$.

Suppose we get that a semantics is a family of networks for evaluation. If we have $\forall M[M \models N_1 \rightarrow M \models N_2]$ we can develop rules for transforming N_1 into N_2 . This is proof theory, LDS style.

Sample problems to be addressed

1. Aggregation of values. A node is attacked and supported by other nodes. How do we aggregate? Joint attacks and disjunctive attacks. Nodes have strength and arcs have transmission values.
2. General theory of attack absorption and its relation to belief change.
3. Temporal dynamics. The effect of temporal change in each network. We need to add a temporal language to the net as well as make all parameters time dependent.

4. Contents and fibring of networks. Giving nodes a content, e.g. a theory, or another network etc.
5. Feedback, e.g. nodes attacking connections.
6. Handling loops.
7. Procedures and theories for propagating values in networks and extracting information from networks.
8. Metalevel, object level of networks, hierarchies.
9. Logic viewed as networks. A logical theory becomes a network.
10. Evaluating one network in another. One is the “theory” and the other is the semantical “model”.
11. Proof theories for networks.

1.2. General view of argumentation networks

Having discussed our general manifesto in the previous subsection, let us outline our view of argumentation networks in general and the place of fibring within this outline.

There are several ways of viewing and handling argumentation networks. Main among them are the logic programming approach, the classical (first-order or higher-order) logic approach and the algebraic equational approach.

For the purpose of fibring networks, the algebraic equational approach is the most convenient.

Let us take as our starting point a classical model (S, R) of a binary relation R on a non-empty set S . This is basically a directed graph. We can read xRy as an arrow from x to y : $x \rightarrow y$. If we view (S, R) as an argumentation network then xRy means x attacks y .

(S, R) in general can be a basis for many types of networks. To give it a more specific character, we need more annotations. Let us add the unary predicates Q_0, \dots, Q_n . So $Q_i(x)$ is some property of x . Specific types of networks will have some axioms relating the relation R to the predicates Q_0, \dots, Q_n .

For example, for argumentation networks we have three predicates Q_0, Q_1, Q_2 , corresponding to the three values $\{in, out, undecided\}$ in a Caminada labelling of arguments, and the relationships between them and R is governed by a theory Δ formalising the Caminada rules. See Definition 2.1 and see [5] for a survey and further results.

In the more general case, we have a theory $\Delta(R, Q_0, \dots, Q_n)$ governing the relationship between R and $\{Q_i\}$. Any model of Δ will be a properly presented network of type Δ .

The kind of questions one can ask in this context is the existence of models of certain types, the existence and nature of models with maximal or minimal Q_i , and relationships between different models.

The mathematical answers we obtain for the above questions will have quality meaning in the context of the particular network we are studying. For example for argumentation network, Q_1 is the set of winning arguments and so the members of Q_1 give the ‘logical content’ of the network. For the general case the above questions are purely mathematical.

We can therefore employ general formal logical techniques to get answers to these questions.

For example, we can use second-order logic to ask for minimal models for Q_0 . We write the wff

$$\theta(Q_0 \text{ minimal}) = \Delta(R, Q_0, \dots, Q_n) \wedge \forall Q'_0, \dots, Q'_n (\Delta(R, Q'_0, \dots, Q'_n) \rightarrow Q_0 \subseteq Q'_0)$$

$(S, R) \models \theta$ says that out of all possible models for Q_i based on (S, R) , our model is the one with Q_0, \dots, Q_n where with Q_0 minimal. To express this we need to quantify over subsets. There are also methods for eliminating second-order set quantifiers or for finding fix point solutions for them. See [16]. These methods can also be profitably employed here, see [18].

First-order logic is not sufficient to express the situation in Figure 1. This is not because the figure contains several types of arrows (attack and support). The different types of arrows give rise to different binary relations. Also the annotations of nodes and arrows is not a problem. We can add parameters to the relations (so unary predicates become binary, binary relations become ternary, etc.). The problem is arrows leading to arrows.

For example in Figure 1 we have an arrow emanating from $c \rightarrow b$ going to the arrow emanating from d and attacking the arrow $a \rightarrow b$.

Writing this in full would yield a relation between 5 elements. This is too complicated to be natural, since these arrows can be iterated to any higher level. Thus a different approach is required if we are seriously dealing with complex networks as displayed in Figure 1.

The second approach is the equational algebraic approach. We regard the predicates $Q_i(x)$ as algebraic values attached to the nodes x . Let $x = a_i$ mean that $Q_i(x)$ holds. We now consider the axioms of Δ as a vehicle using R to formulate a system of equations on the algebra $\mathcal{A} = \{a_i\}$. Of course suitable operations on \mathcal{A} need to be defined. These operations should arise from Δ and respect the declarative content of Δ . To see how this is done for the case of argumentation networks see the discussion in Section 2.2. It is not known what sufficient conditions we can put on a general network of type

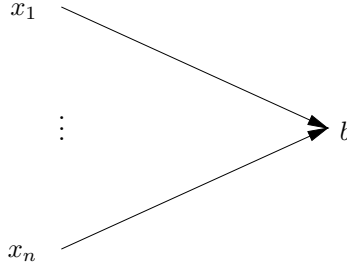


Figure 4.

Δ to ensure that some algebraic equations can be extracted. This approach is better for networks with many levels of arrows attacking arrows. This is because the ‘higher order’ logic part gets processed into the equations and we just end up with complex equations on the nodes. So from this point of view the network is just an instrument for formulating equations.

The third approach is the logic programming approach. In fact in Dung’s original paper [7], the words ‘logic programming’ are in the title. Consider the network in Figure 4. The nodes x_1, \dots, x_n are all the nodes attacking b . The Horn clause corresponding to it is

$$b \text{ if } \bigwedge_i \neg x_i$$

where \neg is negation as failure.

Given a network (S, R) , we translate it into a set of Horn clauses by taking the clauses

$$y \text{ if } \bigwedge_{\{x | xRy\}} \neg x$$

for all $y \in S$.

This translation, originally presented in [15] and further studied in [21] gives a logic program with two special properties.

1. Each literal is the head of at most one clause.
2. All literals in a body of clauses are negated.

Given a logic program with properties (1) and (2) we can regain the corresponding argumentation network by defining

xRy iff (definition) x appears in the body of the clause with head y .

The general question of how to find a corresponding argumentation network for any general logic program, for example

$$\begin{aligned} y &\text{ if } a \wedge \neg b \\ y &\text{ if } c \wedge \neg d \end{aligned}$$

(which does not satisfy (1) and (2) above), is studied in [21]. We use the notion of a critical subset of an argumentation network introduced in Definition 3.3.

Given a general logic program, we can represent it as a classical model of the form $(S, R_1^\pm, \dots, R_k^\pm)$. The elements of S are the literals of the program. The relation xR_i^+y, xR_i^-y mean that the literal x appears positively (resp. negated) in the i th clause with head y .

So we have

$$\text{clause } i \text{ with head } y: y \text{ if } \bigwedge_{xR_i^+y} x \wedge \bigwedge_{zR_i^-y} \neg z$$

1.3. Overview of our approach to fibring

Given a network (S, R) with a node $x \in S$, we want to view it as a variable for which we can substitute values. This question is a logical question and can be asked of any system of any sort provided it allows for atomic elements.

The notion of fibring of networks arises when we substitute for x another network, see Figure 5. There are two options here

- General fibring: we substitute any other network in any network, e.g. substitute a neural network for a node x in argumentation network.
- Self fibring: substitute networks of the same type. So in our case substitute an argumentation network for a node in another argumentation network.

We are faced with two immediate problems

- Give meaning to the substitution
- Generalise the notion of the network so that it is closed under substitution.

We discuss the meaning of the substitution and its properties in Sections 2.1 and 2.2. In section 2.3 we generalise the notion of argumentation network so it is closed under substitution and we study its properties. We call these networks *higher-level networks*.

These are networks with conjunctive and disjunctive attacks. The results in section 2 give rise to methodological considerations and these are studied in Section 3. Our aim in Section 3 is to show the existence of labellings on our new networks. We do this by reducing the new networks to ordinary networks.

Section 4 compares our results with the literature and discusses further research possibilities.

2. Fibering argumentation networks

2.1. The fibering problem

On comparing general logics with networks, we find that certain logical operations, which are basic and widespread in logic, do not occur in networks. One such operation is substitution. Given a logical formula $A(q_1, \dots, q_k)$ built up from the atoms q_1, \dots, q_k we can take q_1 and substitute for it another formula $B(p_j, q_i)$ to obtain $A(B(p_j, q_i), q_2, \dots, q_k)$. The analogous operation in networks of any kind is to take a node q_1 in a network \mathbf{N} and substitute for the node another network \mathbf{N}' . This kind of operation is not naturally done in networks. If we want to integrate logics with networks, it might be helpful if we try to make sense of this operation for the cases of the various networks available such as neural nets, Bayesian nets, argumentation nets, etc.

Figure 5 shows the basic situation.

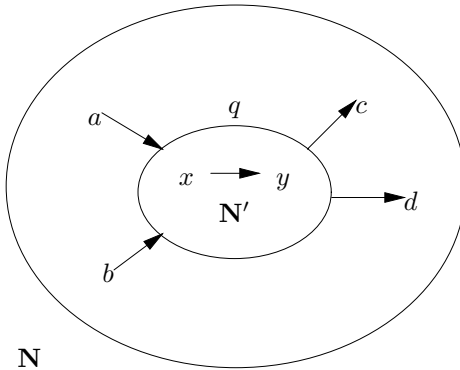


Figure 5.

We substitute \mathbf{N}' for the node q in the network \mathbf{N} , to obtain $\mathbf{N}(q/\mathbf{BN}')$. The node q is connected to the nodes c and d in \mathbf{N} and nodes a and b are

connected to it. The exact nature of the connection is not relevant and it depends on the nature of the network. In \mathbf{N}' we also have internal nodes such as x and y and there is a connection from x to y . These nodes now become nodes in the new network $\mathbf{N}(q/\mathbf{N}')$.

Our problem here is to make reasonable sense of this situation. We need to address the following questions:

Question 1

How do we understand connections from a, b into \mathbf{N}' ? Do they connect in any way to the internal nodes x and/or y ?

Question 2

How do we understand connections emanating from \mathbf{N}' to other points in \mathbf{N} ? Do they emanate from some nodes in \mathbf{N}' ?

Question 3

What do we do with nodes occurring in both \mathbf{N} and \mathbf{N}' ?

These connections were originally going to and from q of \mathbf{N} , but now we have substituted \mathbf{N}' for q , and so we need to answer question 1, 2, and 3. We also have to decide what to do with q itself. Do we leave it in \mathbf{N} or is it out being replaced by \mathbf{N}' ?

Question 4

Some networks have several options internally for what can happen to q . Do we treat the substitution of \mathbf{N}' differently by case analysis depending on what happens to q ? (This is how we do it in Bayesian nets.)

We have already analysed this situation for the case of Bayesian nets and for the case of neural nets. We found natural solutions to the above questions but the solutions vary from network to network. The solution is completely different for neural nets from the case of Bayesian nets. See [6] for a summary.

Question 5

How do we view \mathbf{N}' ? \mathbf{N}' is a network. In the case of Figure 4, \mathbf{N}' is a two point network $x \rightarrow y$. The question we ask is do we ‘process’ network \mathbf{N}'

first (option 1) and then substitute the result, or do we substitute \mathbf{N}' , as is (option 2), first and then process the result. In the case of \mathbf{N}' , processing means taking an extension, so in this case there is only the ground extension, with $x = \text{in}$, and $y = \text{out}$. So if we process \mathbf{N}' first, we ignore y since it is out and substitute only x . Otherwise, we substitute \mathbf{N}' in its entirety. If \mathbf{N}' is the net of Figure 9 then we substitute the set $\{a, c\}$ and we still have to address questions 1–4.

If we choose option 1 and process \mathbf{N}' first we must also address the case where \mathbf{N}' has more than one extension, see for example Figure 17. Do we do case analysis and different substitutions in each case?

We shall address question 5 in the first half of Section 2.3.

We now try to figure out a solution for argumentation nets and see what we can get.

2.2. A fresh look at argumentation networks

We quickly recall some basic definitions in order to present a new point of view on argumentation networks.

DEFINITION 2.1.

1. An argumentation network has the form $\mathbf{P} = (S, R)$ where $S \neq \emptyset$ is the set of arguments and $R \subseteq S^2$ is the attack relation.
2. A Caminada labelling on \mathbf{P} is comprised of three subsets of S , $Q_0, Q_1, Q_2 \subseteq S$ satisfying the following axioms Δ :
 - (a) $\forall x [Q_0(x) \vee Q_1(x) \vee Q_2(x)]$
 - (b) $\sim \exists x [Q_i(x) \wedge Q_j(x)]$ for $i \neq j, i, j = 0, 1, 2$.
 Q_0 is the set of *out* arguments.
 Q_1 is the set of *in* arguments
 Q_2 is the set of *undecided* arguments.
 - (c) $\forall y [\forall x (xRy \rightarrow Q_0(x)) \rightarrow Q_1(y)]$
 - (d) $\forall y [\exists x (xRy \wedge Q_1(x)) \rightarrow Q_0(y)]$
 - (e) $\forall y [(\forall x (xRy \rightarrow (Q_0(x) \vee Q_2(x))) \wedge \exists x (xRy \wedge Q_2(x)) \rightarrow Q_2(y)]$
3. A network may have more than one Caminada labelling to its nodes, which satisfies Δ , see Figure 17. Each such option is called an Extension.
4. $Q_1(x)$ says that x is labelled *in* (or $x = \text{in}$), $Q_0(x)$ says x is labelled *out* (or $x = \text{out}$) and $Q_2(x)$ says that x is undecided (or $x = ?$).

The basic idea of the Caminada labelling can be expressed in the following general terms. Assume x_i are nodes, which we call *units*. (Since these nodes might end up, after substitution, as networks, we prefer to call them units.) These units execute among other things, an attack in the direction of the unit y .

The labelling must satisfy the following conditions:

We use the same numbering as in Definition 2.1, namely: items 2c, 2d and 2e of Definition 2.1.

For the unit y to be *in*, all attacking units x in the direction of y must be *out* as far as the direction of y is concerned.

For the unit y to be *out* it is sufficient that one of its attacking units x is *in* as far as the direction of y is concerned.

2.1.2e For a node to be *undecided* we must have that all the attackers in its direction are *out* or *undecided* with at least one of them being *undecided*.

Argumentation networks (called Argumentation Frames) were introduced by Dung in 1995 through a logic programming point of view. This point of view persists until this day and is adopted in the majority of papers on the subject. Given a network (S, R) as in Definition 2.1, one seeks subsets of S called extensions, satisfying certain fixed point properties. This is parallel to the various extensions of logic programming. Caminada was the first, as far as I know, to present the labelling point of view, as given here in Definition 2.1. However, the Caminada labelling point of view is still tied in with the logic programming extensions point of view.

We need to break away from this point of view and think in terms of labels as functions, giving values to the nodes in some algebraic or numerical range (usually the complex or real numbers). This is the point of view of [2] and [1] and this is what we need for the results of this paper. We are not rejecting or even criticising the logic programming point of view, we simply need the functional point of view to be able to prove some theorems and be able to compare argumentation networks with other networks, following our agenda of unifying logic and networks.

Our point of view is best explained via some examples.

Consider Figure 17. The labelling/extensions point of view will say that this network has three extensions or three Caminada labellings.

1. $a = in, b = out, c = out$
2. $a = out, b = in, c = out$
3. $a = ?, b = ?, c = ?$.

The functional approach will say that we are looking for a numerical or algebraic labelling function $\lambda(q)$ of nodes $q \in S$, giving values in a field of values (the complex numbers will do) satisfying the conditions of Definition 2.1, (written appropriately for λ) as a set of equations.

The conditions to satisfy are:

- (*1) if x_1, \dots, x_n are all the attackers of y then $\lambda(y) = \prod_i (1 - \lambda(x_i))$
- (*2) if y has no attackers then $\lambda(y) = 1$.

To present λ for Figure 17 we use values in $\{0, 1, \frac{1}{2}\}$ or values over the complex numbers.

We get the following system of equations, using (*1) and (*2) to present the equations:

- 1. $\lambda(a) = 1 - \lambda(b)$
- 2. $\lambda(c) = (1 - \lambda(a))(1 - \lambda(b))$

To solve the equations, let $\lambda(a) = t$

We get

- 1. $\lambda(b) = 1 - t$
- 2. $\lambda(c) = (1 - t)t$.

If we let t range over the values $\{0, 1, \frac{1}{2}\}$ we get the Caminada extensions (with $\frac{1}{2} = \text{undecided}$).

We can also use values in an algebra. We define the algebra Cam as follows:

Let $\{in, out, ?\}$ be 3 values and define the operations of inverse and multiplication in Cam as follows:

$$x \mapsto \bar{x} \text{ (we also write } \bar{c} \equiv (1 - x))$$

$$x, y \mapsto x \cdot y$$

$$\bar{in} = out$$

$$\bar{out} = in$$

$$\bar{?} = ?$$

$$in \cdot x = x \cdot in = x$$

$$out \cdot x = x \cdot out = out$$

$$? \cdot ? = ?$$

Note that in Cam , $x^2 = x$, for all x .

Let us now look at Figure 18. Here, we get the equation

$$\begin{aligned} \lambda(a) &= 1 - \lambda(a) \\ \lambda(a) &= \frac{1}{2}. \end{aligned}$$

The difference between the two points of view can be clearly separated by the following three examples, as depicted in Figures 6, 7 and 8.

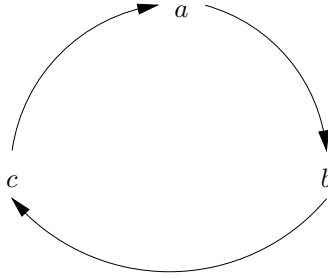


Figure 6.

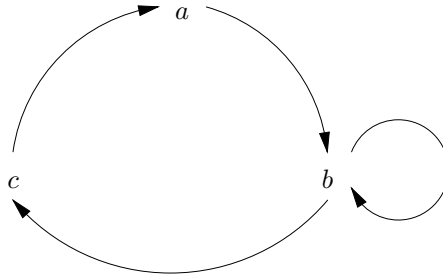


Figure 7.

For Figure 6 we get the equations

1. $\lambda(a) = 1 - \lambda(c)$
2. $\lambda(c) = 1 - \lambda(b)$
3. $\lambda(b) = 1 - \lambda(a)$

The only solution is $\lambda(a) = \lambda(b) = \lambda(c) = \frac{1}{2}$ in the complex numbers and ? in *Cam*.

For Figure 7 we get the equations

1. $\lambda(a) = 1 - \lambda(c)$
2. $\lambda(c) = 1 - \lambda(b)$
3. $\lambda(b) = (1 - \lambda(a))(1 - \lambda(b))$

To solve, we get

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

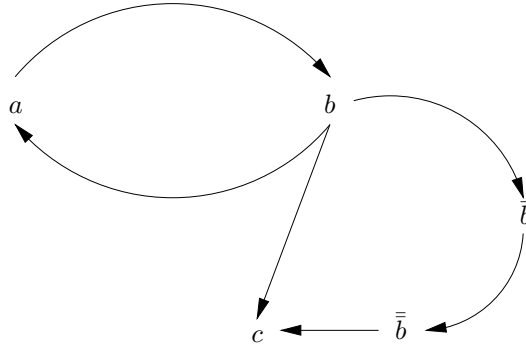


Figure 8.

If our range of values is the algebra *Cam* then we can solve the equations by giving the unique solution

$$\lambda(a) = \lambda(b) = \lambda(c) = ?$$

If our range of values is the complex numbers, we can continue to solve for λ in the complex numbers. We get the equation

$$\lambda(b)^2 - 3\lambda(b) + 1 = 0$$

and solve

$$\lambda(b) = 1.5 \pm \sqrt{1.25}$$

The other values for $\lambda(a)$ and $\lambda(c)$ can be calculated.

The important point is that we can tell the difference between Figure 6 and 7, by choosing the right range for λ .

The Caminada labelling will give the nodes a, b, c value $? = \text{undecided}$ in both cases and will not be able to tell the difference. The argumentation theorist may not care about the difference between the two figures. Both are incoherent. He may concede that the functional point of view may be of interest in comparing with other networks but as far as the area of argumentation itself is concerned, he may believe that we do not need this new point of view.

In fact, however, the functional point of view does make a difference for argumentation theory itself. Figure 8 is the example for that.

Let us write the equations:

1. $\lambda(b) = 1 - \lambda(a)$
2. $\lambda(\bar{b}) = 1 - \lambda(b)$

3. $\lambda(\bar{\bar{b}}) = 1 - \lambda(\bar{b}) = \lambda(b)$.
4. $\lambda(c) = (1 - \lambda(b))(1 - \lambda(\bar{\bar{b}})) = (1 - \lambda(b))(1 - \lambda(b))$.

Note that from (2) and (3) we get that

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

Working in the complex numbers, let $\lambda(a) = t$ and use it as a parameter. Then

1. $\lambda(b) = 1 - t$
2. $\lambda(\bar{b}) = t$
3. $\lambda(\bar{\bar{b}}) = 1 - t$
4. $\lambda(c) = t^2$

Working in *Cam* we observe the following when comparing Figure 8 with Figure 17.

Ignoring the nodes \bar{b} and $\bar{\bar{b}}$, we get that Figure 17 is a subnetwork of Figure 8. In fact the equations we get for Figure 8 for the nodes a, b, c would be the same equations as those of Figure 8 provided that $(1 - \lambda(b)) \cdot (1 - \lambda(b)) = (1 - \lambda(b))$ holds. Indeed, this equation does hold for *Cam*. So as far as *Cam* is concerned Figure 8 is a conservative extension of Figure 17. Any λ on $\{a, b, c\}$ can be uniquely extended to a λ^* on Figure 8 in a consistent manner. The equations for Figure 8 give the same solutions to the nodes of Figure 17 as the solutions to the equations of Figure 17.

Let us consider another example, that of Figure 50.

We get the following equations:

1. $\lambda(\bar{x}) = 1 - \lambda(x)$
2. $\lambda(\bar{a}) = 1 - \lambda(a)$
3. $\lambda(\bar{c}) = 1 - \lambda(c)$
4. $\lambda(e(x, a)) = (1 - \lambda(\bar{a}))(1 - \lambda(\bar{x})) = \lambda(a)\lambda(x)$
5. $\lambda(e(x, c)) = (1 - \lambda(\bar{x}))(1 - \lambda(\bar{c})) = \lambda(x)\lambda(c)$
6. $\lambda(\bar{\bar{a}}) = 1 - \lambda(\bar{a})$
7. $\lambda(\bar{\bar{c}}) = 1 - \lambda(\bar{c})$
8. $\lambda(e(x)) = (1 - \lambda(\bar{\bar{a}}))(1 - \lambda(\bar{\bar{c}}))(1 - \lambda(\bar{x}))$
9. $\lambda(a) = (1 - \lambda(e(x)))(1 - \lambda(e(x, c)))$
10. $\lambda(c) = (1 - \lambda(e(x)))(1 - \lambda(e(x, a)))$.

Simplifying, we get

4. $\lambda(e(x, a)) = \lambda(a) \cdot \lambda(x)$
5. $\lambda(e(x, c)) = \lambda(c) \cdot \lambda(x)$
6. $\lambda(\bar{a}) = \lambda(a)$
7. $\lambda(\bar{c}) = \lambda(c)$
8. $\lambda(e(x)) = (1 - \lambda(a))(1 - \lambda(c))\lambda(x)$
9. $\lambda(a) = [1 - [\lambda(x)(1 - \lambda(a))(1 - \lambda(c))]] \cdot [1 - \lambda(x)\lambda(c)]$
10. $\lambda(c) = [1 - [\lambda(x)(1 - \lambda(a))(1 - \lambda(c))]] \cdot [1 - \lambda(x)\lambda(a)]$

Let us, for example, see what we get if we substitute in the equations the values $x = ?$ $a = in$ and $c = ?$.

First note that from equations (1)–(8) we get that the values for $\bar{x}, \bar{a}, \bar{c}$, $e(x, a)$, $e(x, c)$, \bar{a} , \bar{c} and $e(x)$ are uniquely determined. The problem we face is whether our choice of the above values for $\{x, a, c\}$ is fortunate so that equations (9) and (10) also hold.

Let us check. We get

1. $\bar{x} = ?$
2. $\bar{a} = out$
3. $\bar{c} = ?$
4. $e(x, a) = (1 - out)(1 - ?) = in \cdot ? = ?$
5. $e(x, c) = (1 - ?)(1 - ?) = ? \cdot ? = ?$
6. $\bar{a} = in$
7. $\bar{c} = ?$
8. $e(x) = (1 - in)(1 - ?)(1 - ?) = out \cdot ? = out$
9. $a = (1 - out)(1 - ?) = in \cdot ? = ?$
10. $c = (1 - out)(1 - ?) = in \cdot ? = ?$

We get that equation 9 is contradictory. So we guessed wrong!

2.3. Joint and disjunctive attacks

We are faced with the need to make sense of situation illustrated in Figure 9 as an example.

We got Figure 9 by substituting the network of Figure 10 for the position q in the network of Figure 11.

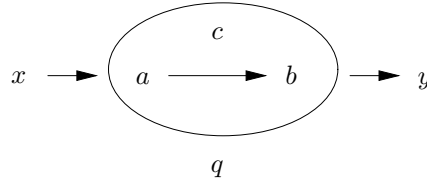


Figure 9.

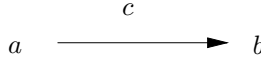


Figure 10.

We need to answer Questions 1, 2, 3, 4 and 5 for this case. It is a simple case since the substituted network of Figure 10 has no loops and so it has a clear message, only one extension: a is *in*, c is *in* and b is *out*. So effectively b is *out* and plays no active ‘in’ role in the network.

We must clarify our concepts at this junction: Figure 9 shows a substitution of the network of Figure 10 at position q of Figure 11.

The first choice of options we have to do is to ask Question 5.

- (*1) Do we process the network of Figure 10 first, i.e. choose an extension for it and then and only then substitute the result for q in Figure 11? (Call this Option 1.),

or

- (*2) We substitute Figure 10 as is, as a network, and define whatever is supposed to happen (call this Option 2).

If we follow option 1 for the network of Figure 10, we get the network of Figure 12. It is as if we substitute the set $\{a, c\}$ for the node q in Figure 11.

We begin by examining Option 1 and study the formal network of Figure 12. We need to decide how to define the concepts of attacks (see Figure 13). Figure 12 has two problematic parts, $\{a, c\}$ attacking y and x attacking $\{a, c\}$, as shown in Figure 13.

- (*3) What does it mean for a unit (argument) x to attack a set $\{a, c\}$?
- (*4) What does it mean for a set $\{a, c\}$ to attack a unit (argument) y ?

The case (*4) is easy. For $\{a, c\}$ to be *in* we must have both a and c *in*. In this case y is *out*. See Figures 13 and 14.

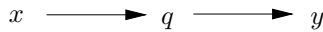


Figure 11.

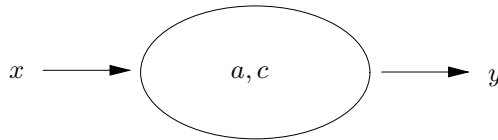


Figure 12.

For the case of (*3), when the unit $\{a, c\}$ is attacked by x , we have two possibilities:

Possibility 1 for (*3)

x must attack one of the elements of the set i.e. either x attacks a or x attacks b .

Possibility 2 for (*3)

x attacks the “in” status of $\{a, b\}$, that is we need to have either a out or b out.

Possibility 2 is the symmetrical counterpart to (*4). For $\{a, c\}$ to be *in* we must have “ a is *in* and b is *in*”, therefore for $\{a, c\}$ to be *out* we must have “ a is *out* or c is *out*”.

The two possibilities are not equivalent.

This will be discussed in detail at the beginning of Section 3 and in Section 4.2. We shall see later that Nielsen and Parsons [20, 19] proposed joint attacks, which is coming from a completely different point of view, follows possibility 1. We shall compare our approaches in Section 4.2.

DEFINITION 2.2 (Joint and disjunctive attacks). We adopt possibility 2 for (*3). So the definition is as follows (see Figure 15).

- x attacks $\{e_1, \dots, e_m\}$ (disjunctively) means:
 $x = in$ implies $\bigvee_{i=1}^m e_i = out$
 (especially this can mean that if x is *in* then several or more e_i are *out*).
- $\{e_1, \dots, e_m\}$ (jointly) attack y means:
 $\bigwedge_{i=1}^m e_i = in$ implies y is *out*.
- $\{e_1, \dots, e_m\}$ is *in* iff $\bigwedge_{i=1}^m e_i = in$
- $\{e_1, \dots, e_m\}$ is *out* iff $\bigvee_{i=1}^m e_i = out$

EXAMPLE 2.3 (Caminada–Gabbay labelling for joint and disjunctive attacks). We illustrate/define our labelling using the typical case of Figure 15.

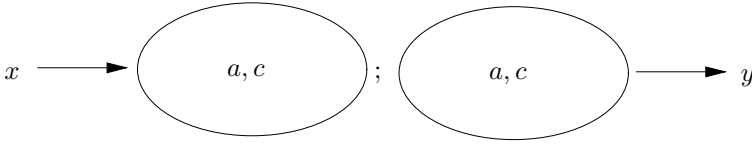


Figure 13.

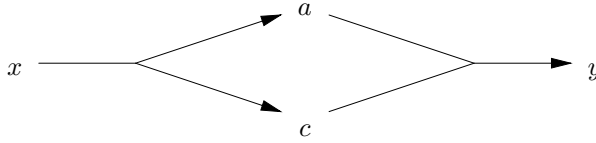


Figure 14.

1. If x is labelled *in* and $e_j, j = 1, \dots, k$ are the nodes being disjunctively attacked from x then at least one of e_j must be *out*.
2. If y is jointly attacked from $e_j, j = 1, \dots, k$ and all of e_j are *in* then y must be *out*.
3. Suppose q is targeted by a disjunctive attack from x_1, \dots, x_n . How can q be *undecided*? To see what must be done consider Figure 16 showing a typical situation, where q is attacked from two different directions.

In Figure 16, q is being attacked from two different directions. One attack is emanating from x and one attack emanating from y . The attack emanating from x involves also v_1 and v_2 which are also being disjunctively attacked and the attack emanating from y involves also w_1 and w_2 . We imagine Figure 16 as a subnetwork of a larger network \mathbf{N} . So it is quite possible that x or v_i or w_j are being attacked from other parts of the network. We assume that q is being attacked only from x and y .

Let us consider an arbitrary attack on q , emanating from a node z and involving nodes u_1, \dots, u_k which are also attacked. Thus to be explicit, z disjunctively attacks $\{u_1, \dots, u_k, q\}$.

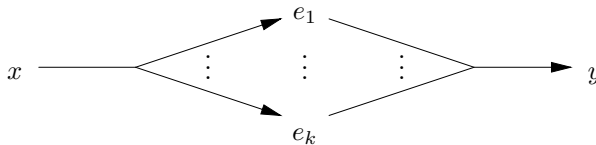


Figure 15.

Let us use the terminology that q is subject to an attack emanating from z . We now want to define three concepts describing this attack.

- (#1) the attack on q emanating from z is not a threat to q .
- (#2) the attack from q emanating from z forces q *out*
- (#3) the attack on q emanating from z makes q *undecided*.

Let us define these three concepts:

- (#1) This attack is not going to be a threat to q if one of u_1, \dots, u_k is *out*. Say u_i is *out* because of an attack on it from the rest of the network \mathbf{N} , in which Figure 16 is embedded. If one of the u_i is *out* then the attack succeeds without ‘hitting’ q . So this attack is no threat to q and q can have any value, *in*, *out* or *undecided*.
- (#2) the attack on q from z forces q *out* if $z = \textit{in}$ and all of u_1, \dots, u_k are *in*.

- (#3) when do we say that the attack on q from the direction of z makes q *undecided*? There are two cases:

- (a) z itself is *undecided*. So we do not know whether z is *in* or *out*. If some u_i is *out* then it does not matter that z is *undecided*. q is not threatened. If all of u_i are not *out*, but are either *in* or *undecided*, then there is a real possibility of attack on q . If z and all of u_i turns out to be *in* then q must be *out* and if one of $\{z, u_1, \dots, u_k\}$ is *out* then q can be *in*. Thus we must make q *undecided*.

Now we ask can u_1 be *in*? Well, it cannot be *in* because from the point of view of u_1 , we see the same situation as that which we saw from q . We see $z = ?$ and u_1, \dots, u_k and q are not *out*. So u_1 must also be *undecided*. So to maintain coherence of our rules we must declare (*) below:

- (*) If a disjunctive attack emanates from z in the direction of u_1, \dots, u_k, q and z is *undecided* and none of u_1, \dots, u_k, q is *out* (u_1, \dots, u_k, q can be *out* owing to attacks from the rest of the network) then u_1, \dots, u_k, q are all *undecided*. Let us refer to this situation as u_i, q become *undecided* because of a disjunctive attack on it emanating from an undecided z .
- (b) Let us now check what happens if $z = \textit{in}$. Can we still get $q = \textit{undecided}$? The answer is yes. Suppose u_1, \dots, u_k are all attacked from some respective z_1, \dots, z_k which are all *undecided*. This will make u_1, \dots, u_k *undecided* according to (*) above.

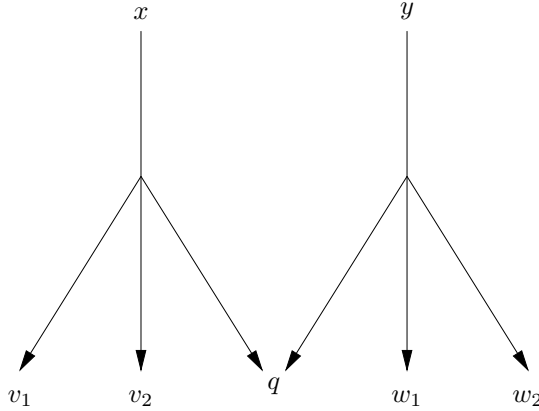


Figure 16.

So if $u_1 \dots$ or $\dots u_k$ ends up *out*, then q is safe, and if all of u_1, \dots, u_k end up *in*, then q has to be *out*. Thus we must declare q undecided. So we get rule (**) below:

- (**) If a disjunctive attack emanates from z in the direction of u_1, \dots, u_k, q and at least one of u_i is undecided (because for example of a disjunctive attack on u_i emanating from an undecided z_i) then all of u_1, \dots, u_k, q are all undecided.

We can now define what it means for q to be undecided, by using (*) and (**):

- (***) q is undecided if
- (1) q is not forced out by any attack on it emanating from any z .
 - (2) q can be shown undecided by repeated applications of (*) and (**) above.

Let us now check when v_1 can be undecided in Figure 16 v_1 can be undecided in Figure 16 if for example x is undecided and there are no attacks that force any of v_1, v_2 or q *out*.

v_1 can be undecided if $x = \textit{in}$ but y is undecided and there are no attacks forcing any of v_1, v_2, q, w_1, w_2 *out*.

Now having adopted our notion of disjunctive and joint attacks in Definition 2.2, we are ready to discuss further whether to adopt Option 1 for network substitution, namely whether to choose an extension for the network first before we substitute and then substitute only the nodes that are *in* in that extension. This discussion will be somewhat lengthy (terminating

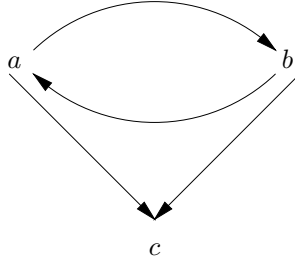


Figure 17.

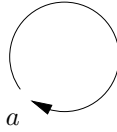


Figure 18.

just before Definition 2.4) and will result in rejecting option 1 in favour of option 2. The weakness of option 1 is that it gets complex in the presence of loops.

The above considerations explained what happens with option 1 when the substituted network has no loops so it has no undecided nodes and therefore it has only one extension. What do we do when the substituted network has loops? There are two typical cases to consider, as shown in Figures 17, 18.

Figure 20 is obtained by substituting Figure 17 into Figure 19 at node q and Figure 21 is obtained by substituting Figure 18 into Figure 19, at node q .

In the first case we have two options for the network, the extensions $\{a, c\}$ and $\{b, c\}$. Which one do we take? This is not really a problem because we can push the problem onto the labelling. So depending on the labelling we get either Figure 19 or Figure 23 we still have the problem of how do we represent the original network?

Let us try the representation as in Figure 24.

If a is *in* and c is *in*, we have a case of $\{a, c\}$ and so since x must be *in*, its attack on $\{a, c\}$ must take one of $\{a, c\}$ out. Similarly if b is *in* and

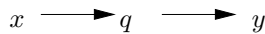


Figure 19.

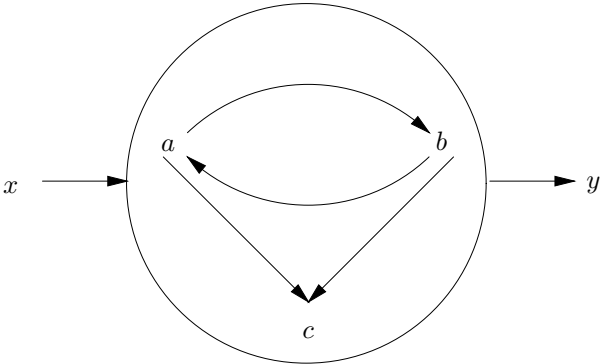


Figure 20.

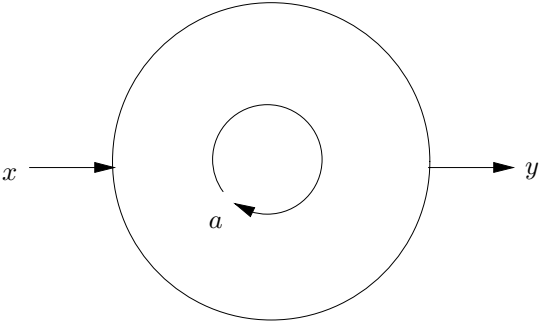


Figure 21.

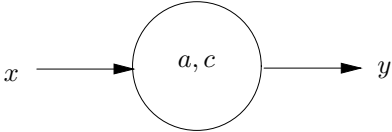


Figure 22.

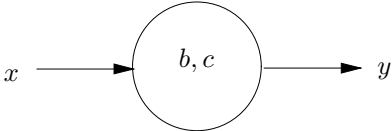


Figure 23.

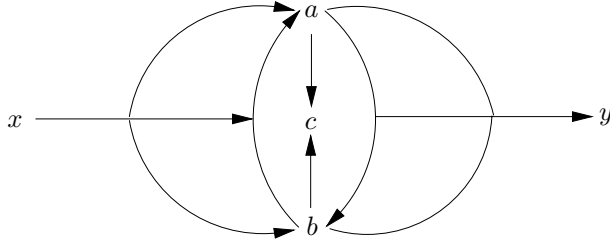


Figure 24.

c is *in*, we have a case of $\{b, c\}$ and so since x must be *in*, its attack on $\{b, c\}$ must take one of $\{b, c\}$ out.

We immediately see that we have a problem with the proposed representation of Figure 24. In the figure we put in the entire network of Figure 17 into Figure 19. Thus when b is *in*, a is *out* and when a is *in* we have that b is out. So formally, for the case let us say, that $\{a, c\}$ are *in* and b is out, we set that the attack from x is formally successful, because in our diagrams for this case b is *out* so $\{a, c\}$ can stay *in*, contrary to our intentions! Similarly, since always either a or b is *out* then the substituted network can never attack anything as represented in the diagram of Figure 24 because formally the joint attack from $\{a, b, c\}$ always has one node out (either a or b). Obviously, we need first to calculate the result for the substituted network of Figure 17 and get say that $\{a, c\}$ are *in* and b is *out* and only afterwards address the attack from x and the attack on y .

But then in this case we must make it clear that b , being *out* does not play a role in the considerations of these attacks.

‘Making things clear’ means additional labels. Maybe several kinds of annotated ‘*out*’ with information when each ‘*out*’ is to be taken into account.

The labelling and the substitutions become hierarchical.

This is not good, not only because of complexity considerations but also conceptually. When we substitute we allow ourselves to use any arguments, so this b in question could be x . Now what do we do? In the hierarchical evaluation the $b = x$ inside Figure 17 as substituted is *out* while the $b = x$ originally outside the figure is *in*. But they are the same $b = x$! So what do we do?

Let us look at Figure 21 and see if we can get a clue as to what to do. Figure 18, as substituted into Figure 19 is both attacking (Figure 25) and is being attacked (Figure 26). So what do we do with Figures 25 and 26?

The latter, Figure 26 is clear, x can take a *out*.

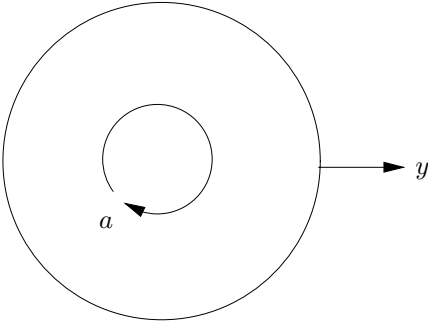


Figure 25.

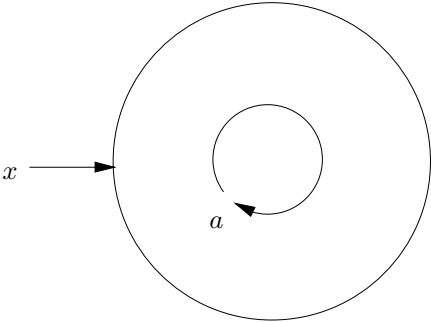


Figure 26.

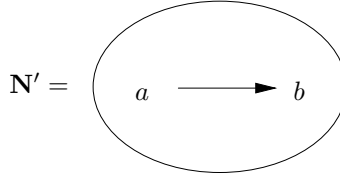


Figure 27.

On the other hand, we look at Figure 25, then since a is *undecided*, we must give y *undecided*. Now we can see that the proposed hierarchical evaluation, when applied to Figure 21 is no good. Evaluating hierarchically will make a *undecided*, y *undecided* and x *in* which is not a good solution for Figure 21. However, evaluating directly without any hierarchical considerations gives us that x is *in*, since it is not attacked, a is *out* and y is *in*. All nice and clear.

The above discussion shows that we had better reject Option 1 and view any substituted network N' into another network N at a point $q \in N$, in which q attacks another node $y \in N$, as a joint attack from *all* the arguments of N' . If some of them are *out* in N' then we regard N' as not conflict-free. Thus the network of Figure 27 is not conflict-free.

The reason we take this view is because when we substitute N' of Figure 27 into another network N , b and a may be in N in other places and we do not know what can happen, a may end up *out* or b *in*.

The above is our Option 2 which we shall adopt.

We now have an agreed sequence of definitions.

DEFINITION 2.4 (Higher level networks).

1. Let $S \neq \emptyset$ be a set of nodes. Let S^0 be the family of all finite non-empty subsets of S , identify the singleton $\{x\}$ with x for simplicity of notation.
2. A higher level argumentation network has the form (S, S^0, R) where S and S^0 are as above and $R \subseteq S^0 \times S^0$ is the attack relation. XRY represents a joint attack from the set X disjunctively attacking the set Y . When $X = \{x\}$ and $Y = \{y\}$ we get an ordinary point to point attack and so when $R \subseteq S \times S$ we get the usual network definition. We represent the situation as in Figure 28

$$\begin{aligned} X &= \{x_1, \dots, x_n\}. \\ Y &= \{y_1, \dots, y_n\}. \end{aligned}$$

or perhaps Figure 29 is more clear.

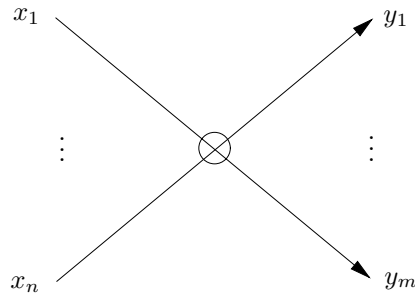


Figure 28.

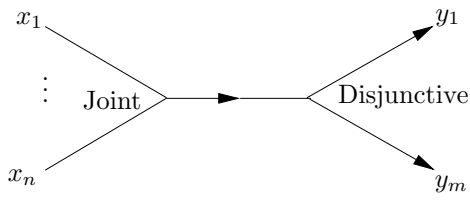


Figure 29.

3. We understand Figure 28 as saying that the set $\{x_1, \dots, x_n\}$ is jointly mounting a disjunctive attack on the set $\{y_1, \dots, y_m\}$. So only if all the x_i are *in* can the attack go forward and in which case we expect at least one of the y_j to be *out*.

We now want to define substitution of one such higher level network into another. The result will again be a higher level network.

We will understand better how to define the substitution after we do some examples.

EXAMPLE 2.5 (Network substitution). Start with a simple network of Figure 30.



Figure 30.

Now substitute Figure 31 for y and substitute Figure 32 for x and get Figure 33.

Figure 33 should be written as Figure 34.

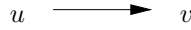


Figure 31.



Figure 32.

Now substitute Figure 35 for u in Figure 34 and get Figure 36.

Note that in Figure 36 we have a joint attack of a and ' $a \rightarrow x$ ' on some target. According to our Option 2, we regard this as a joint attack from $\{a\} \cup \{a, x\}$ on the target, i.e. a joint attack from $\{a, x\}$.

Of course Figure 36 needs to be simplified since a is written twice. We get Figure 37.

We numbered the attacks in Figures 36 and 37 so it will be clear what attack in Figure 36 became what attack in Figure 37.

We try as an exercise to substitute for u in Figure 34, a new figure, Figure 38. So instead of substituting Figure 31 as we did above, we substitute Figure 38.

In this case we get Figure 39.

Now this figure needs to be simplified to Figure 40. In Figure 40 we have:

- 1 is a joint attack of $\{a, v\}$ on v
- 2 is a joint attack of $\{a, v\}$ on a
- 3 is a joint attack of $\{a, v\}$ which disjunctively targets a and v .

REMARK 2.6. We can see that the graphs can get very complex. We note, however, that we cannot get everything, just by repeated substitution. For example, we believe we cannot get Figure 41 by mere substitutions:

We cannot get that x sends a disjunctive attack in the direction of a and in the direction of c but the attack disjunct going towards c joins forces and becomes conjunctive with the attack from b on c .

The only possibilities we can generate are Figures 42 and 43.

Figure 41 is perfectly OK, but I don't think we can get it by repeated substitution of ordinary networks. Figure 41 has three extensions:

- 1. $x = in, a = out, b = in, c = out$
- 2. $x = in, a = in, b = in, c = out$
- 3. $x = in, a = out, b = in, c = in$.

It is debatable whether extensions (2) and (3) are acceptable. This depends on how we understand Figure 41. See section 4 for a discussion.



Figure 33.

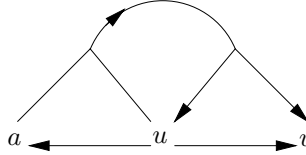


Figure 34.

We must bear in mind that we are not just defining generalisations of argumentation networks from a mathematical point of view. We are following the methodological manifesto of Section 1 and there are good reasons for substituting one network in another. We asked ourselves in Section 2 what general notion of networks we need to allow for substitution. That is when we substitute such a network in another such network, we get a result of the same kind. We found that we need the notion of higher level network of Definition 2.4. So generalising the definition of higher level networks, to allow for Figure 41 requires independent conceptual justification. Substitution is justified intuitively. Joint attacks exists in common sense arguments. Attacking joint attacks also makes sense and this gives rise to disjunctive attacks. How can the mathematical situation of Figure 41 be justified or explained?

We can intuitively say that the disjunctive attack arises from an attack on a joint unit such as x attack on $\{a, c\}$ in Figure 13. To be successful, x wants either a or c to be *out*. So why not joint forces with some b to attack c and increase his chances? This makes sense. We get Figure 41. x will not insist on $c = \text{out}$ if his attack on a succeeds. In Figure 43, $\{x, b\}$ insist on $c = \text{out}$ even if $a = \text{out}$. In Figure 43, $\{x, b\}$ attack on c is not part of a joint attack.

Let us postpone this to the discussion to Section 4.

We can now define higher level substitution.



Figure 35.

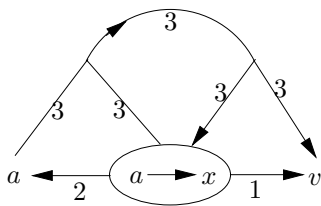


Figure 36.

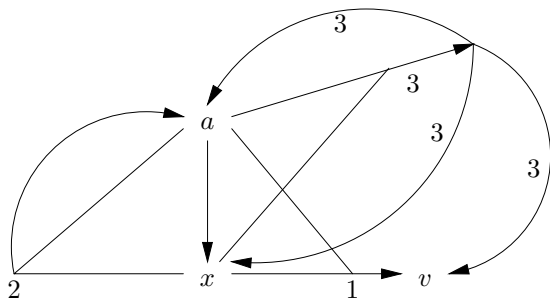


Figure 37.



Figure 38.

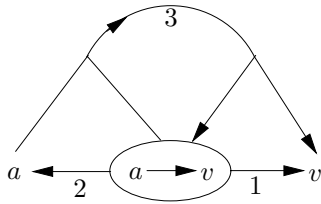


Figure 39.

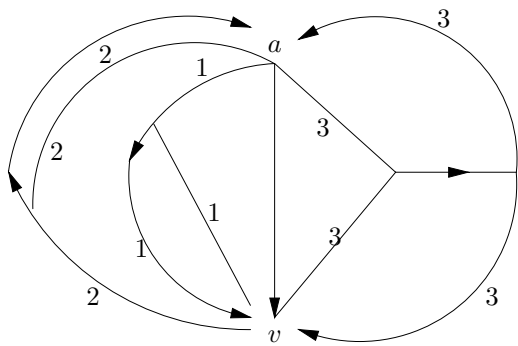


Figure 40.

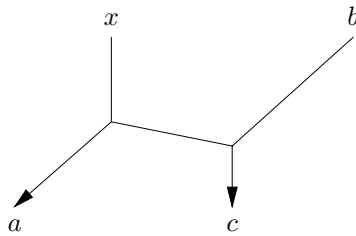


Figure 41.

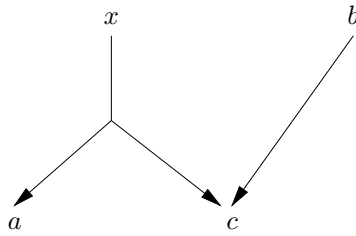


Figure 42.

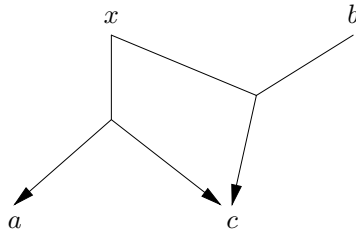


Figure 43.

DEFINITION 2.7. Let $\mathbf{P}_i = (S_i, S_i^0, R_i)$ for $i = 1, 2$ be two *finite* higher level networks. Let $x \in S_1$ be a node. We want to define the network

$$\mathbf{P} = \mathbf{P}_1(x/\mathbf{P}_2)$$

being the result of the substitution of \mathbf{P}_2 for x in \mathbf{P}_1 .

The set of points of \mathbf{P} is $S = S_1 \cup S_2$. We need to define $R \subseteq S^0 \times S^0$. We shall follow the traditional practice used in substitution in logic and assume that x itself is not present in S_2 .

We therefore have two types of available relations.

1. Type 1 from \mathbf{P}_1 :

$$\{a_1, \dots, a_m, x\} R_1 \{b_1, \dots, b_k, x\}$$

where x may not appear among the a_i or not appear among the b_j . In this case we take

$$\{a_1, \dots, a_m\} \cup S_2 R \{b_1, \dots, b_k\} \cup S_2$$

where S_2 will appear wherever x appears.

2. Type 2 from \mathbf{P}_2

$$X R_2 Y$$

in which case we take $X R Y$.

We have thus really defined R to be

$$R =_{\text{def}} R_2 \cup R_1(x/S_2)$$

with the understanding that $\{e_j, S_2\} = \{e_j\} \cup S_2$.

EXAMPLE 2.8. Let us examine again the network in Figure 41. We claimed that such a network cannot arise in our definition of higher level networks.

Since clearly a and c are being attacked and x and b are the attacking nodes, then only the following higher level attacks are possible in the form $X R Y$, as shown in Figure 45.

Figure 45 shows the connections of Figure 44 using points as nodes. It is clear that the attack pattern of Figure 41 is not present in Figure 45.

DEFINITION 2.9 (Caminada–Gabbay labelling). Let $\mathbf{P} = (S, S^0, R)$ be a higher order network.

Let λ be a function giving values in $\{0, 1, ?\}$ to each $s \in S$.

We say this function is a proper labelling iff the following holds.

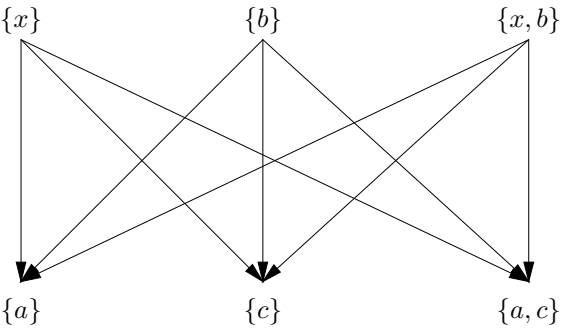


Figure 44.

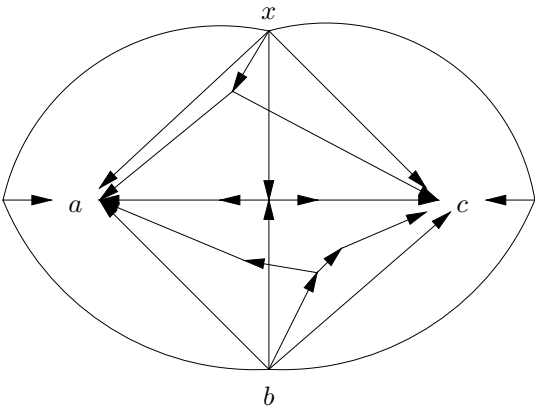


Figure 45.

1. If $XR Y$ holds and $\forall x \in X(\lambda(x) = 1)$ then $\exists y \in Y(\lambda(y) = 0)$.
2. If for every X, Y such that $y \in Y$ and $XR Y$ holds we have that $\exists x \in X$ s.t. $\lambda(x) = 0$ then $\lambda(y) = 1$.
3. If for all X, Y such that $y \notin X \cup Y$ and $XR Y \cup \{y\}$, we have that $\forall z \in X \cup Y(\lambda(z) \in \{1, ?\})$ and $\exists z \in X \cup Y(\lambda(z) = ?)$. Then $\lambda(y) = ?$.

REMARK 2.10. *Figure 46. If $\lambda(x) = 1$ and $\lambda(a) = 0$ then $\lambda(c)$ can be either 1 or 0. Both cases are acceptable. If Figure 46 is part of a larger network and because of attacks from the larger network either $\lambda(x) = ?$ then we cannot have $\lambda(a) = 1$ and $\lambda(c) = ?$.*

3. Methodological results

We are now in a position where we have a definition of a new kind of argumentation network, (S, S^0, R) as defined in Definition 2.4, where sets of arguments can attack sets of arguments. Never mind how this definition was motivated. We got such networks because we wanted our networks to be closed under fibring/substitution (Definition 2.7). The question now is, what are the extensions of this new kind of network? We need to do Dung style analysis of extensions, or develop some other means of answering this question.

Given $E \subseteq S$, is E an extension? Let us see what we need to do Dung style; something like the following:

The attacks are done by sets of arguments on sets of arguments. So we need to define the notion of a conflict free family \mathbb{E} of sets of arguments. Then we define the notion of a set X of arguments being acceptable to \mathbb{E} . Then we define a function φ

$$\varphi(\mathbb{E}) = \{X \mid X \text{ is acceptable to } \mathbb{E}\}$$

Then we say that \mathbb{E} is admissible if $\mathbb{E} \subseteq \varphi(\mathbb{E})$ and now we can define extensions, e.g. \mathbb{E} is complete iff $\mathbb{E} = \varphi(\mathbb{E})$. \mathbb{E} is a family of subsets of the set of arguments S . From \mathbb{E} maybe we get the extensions say

$$E = \bigcup_{X \in \mathbb{E}} X$$

At this point we do not know if this can work or not. We need to try it and develop the correct definitions by trial and error. We are not going to do that. We are going to take another route. We are going to start with (S, S^0, R) and add new points S^* to S and get a new network $S \cup S^*$.

We define an ordinary R^* on $S \cup S^*$ and then embed our network (S, S^0, R) into the network $(S \cup S^*, R^*)$ by implementing the idea of joint and disjunctive attacks using the additional points of S^* .

The new network $(S \cup S^*, R^*)$ is an ordinary Dung network, so it has extensions. Let E^* be an extension. We will define any $E = E^* \cap S$, as an extension of (S, S^0, R) . For this to work properly we need the following critical property:

Critical property of S in $S \cup S^*$

Let λ_1^* and λ_2^* be two Caminada labelling on $S \cup S^*$. Assume that $\lambda_1^* \upharpoonright S = \lambda_2^* \upharpoonright S$, then $\lambda_1 = \lambda_2$.

The critical property of S in $S \cup S^*$ ensures the embedding is faithful!

So now that we know what we are doing, let's roll!

We now reduce higher level networks to ordinary networks. We do this in several stages.

1. Reduce the disjunctive attacks to joint attacks.
2. Reduce the joint attacks to single attacks.
3. Derive the existence of labellings and extensions from (1) and (2).

3.1. Conceptual analysis of disjunctive attacks

Before we embark on any reductions, we must fully clarify the properties of disjunctive attacks.

Consider the disjunctive attack part of Figure 13. We have the following situation (see Figure 46):

- $x = in$ implies $a = out$ or $c = out$

The following are the three possibilities

1. $x = in, a = in, c = out$
2. $x = in, a = out, c = in$
3. $x = in, a = out, c = out$

Figure 46 cannot be reduced to Figure 47 which is properly written as Figure 48.

Figure 48 does not allow for the extension

4. $x = in, a = out, c = out$

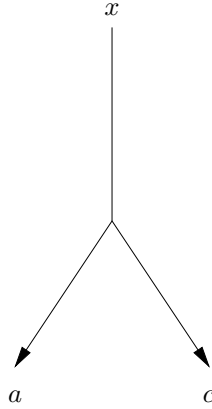


Figure 46.

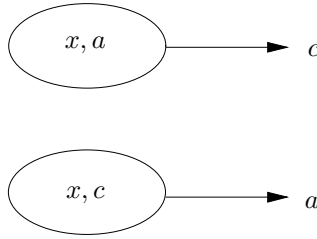


Figure 47.

The meaning of Figure 46 is a disjunctive attack on $\{a, c\}$ and not any specific attack on either a or c . This is also reflected in the problems we had in interpreting Figure 41.

We have three reading of Figure 46.

- (r1) x attacks the coalition (joint) of $\{a, c\}$. x would be happy to see either $a = out$ or $c = out$ but is not mounting any specific attacks on a or on c . According to this interpretation Figure 41 does not have meaning. b cannot join any attack from x onto c . There is no such attack.
 - (r2) x sends two attacks, one in the direction of a and one in the direction of c . x is happy if at least one of them is successful, but would also be happy with both successful.
 - (r3) x sends two attacks as in (r2) above but does not want both to succeed, only one.
- (r1) and (r2) above say:

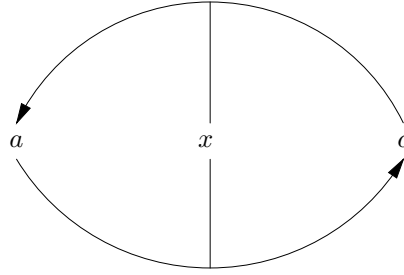


Figure 48.

- $x = in$ implies $a = out$ or $c = out$

while (r3) says:

- $x = in$ implies $(a = out \wedge c = in)$ or $(a = in \text{ and } c = out)$.

Figure 48 corresponds to (r3).

Note that we need not make a choice between (r1) and (r2) for the purpose of this section. We shall see in Section 4.2 that for the purpose of explaining Figure 41 we need to adopt (r2).

How can we represent (1) or (2)? What is the corresponding figure?

We need auxiliary points. Consider Figure 49:

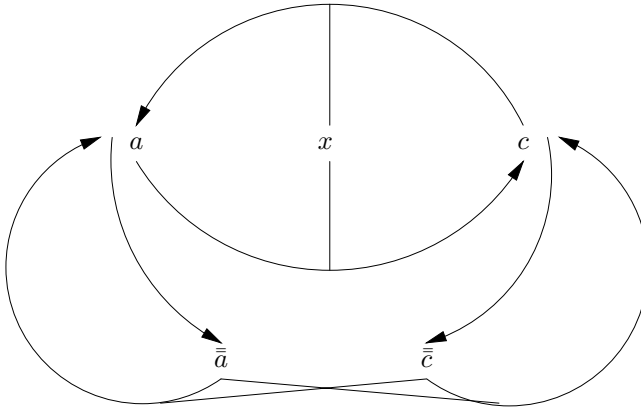


Figure 49.

Figure 49 extends Figure 48 by adding two joint attacks from $\{a = out, c = out\}$ one on a and one on c . To do this properly we use the help of two new intermediary points \bar{a} and \bar{c} . When $a = out$ and $c = out$, we get $\bar{a} = in$ and $\bar{c} = in$ and $\{\bar{a}, \bar{c}\}$ mount the joint attacks on a and c .

Let us see what happens in Figure 49. Since x is not attacked, $x = in$. We now consider our possibilities for a and c . If at least one of $\{a, c\}$ is *in*, the joint attacks of $\{\bar{a}, \bar{c}\}$ fail and we are back to Figure 48. Figure 48 behaves as we want except in the case of $a = out$ and $c = out$. But in this case, $\bar{a} = \bar{b} = in$ and so the joint attacks of $\{\bar{a}, \bar{c}\}$ on a and c must succeed, thus confirming the assumption that $a = b = out$.

We can display the situation without joint attacks in Figure 50.

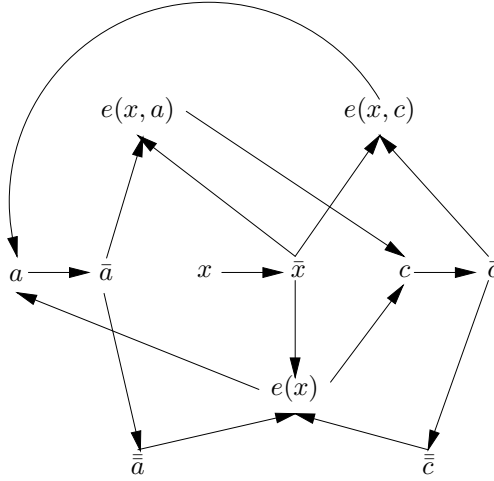


Figure 50.

We used auxiliary points as follows:

1. With each node involved, x, a, c we added new nodes $\bar{x}, \bar{a}, \bar{c}$ and $\bar{\bar{a}}, \bar{\bar{c}}$.
2. We added the intermediaries $e(x), e(x, c), e(x, a)$.

Nodes x, a, c attack only via $\bar{x}, \bar{a}, \bar{c}$ respectively.

$e(x, a)$ is an intermediary doing the job of (x, a) attack on c . Similarly $e(x, c)$ is an intermediary for representing the attack of (x, c) on a .

These two represent the situation of Figure 48. To get also the option of $x = in, a = out, c = out$, we use $\bar{\bar{a}}, \bar{\bar{c}}$ which attack $e(x)$. $\bar{\bar{a}}$ attacks \bar{a} and $\bar{\bar{c}}$ attacks \bar{c} .

EXAMPLE 3.1 (Analysis of Figure 50). Figure 50 can be stand-alone or can be embedded inside a larger Figure. If Figure 50 is stand-alone, then x is the only node which is not attacked by anything. Hence $x = in$. Therefore $\bar{x} = out$. We now examine four options for a and c :

1. $a = out, c = out$
2. $a = in, c = out$
3. $a = out, c = in$
4. $a = in, c = in$

We check consistency of the *in-out* labelling, for the case $x = 1$.

Case 1

If $a = c = out$, we get $\bar{a} = \bar{c} = in$. Therefore $e(x, a) = e(x, c) = \bar{a} = \bar{c} = out$. Also $e(x) = in$.

$e(x)$ attacks a and c confirming they should be out.

So (1) is a consistent labelling.

Case 2

If $a = in$ and $c = out$ then $\bar{a} = out$ and $\bar{c} = in$. Hence $e(x, a) = in$ (it has not attackers, both \bar{x} and \bar{a} are *out*) and $e(x, c)$ is *out*.

Since $e(x, a) = in$ and $e(x, a)$ attacks c it confirms c is *out*.

Since \bar{a} is *out* we get \bar{a} is *in*, therefore $e(x)$ is *out*. Hence a is not attacked by anything (both $e(x)$ and $e(x, c)$ are *out*), confirming that a is *in*.

We also get \bar{c} is *out* in \bar{c} is *in*. This does not affect $e(x)$, because \bar{a} is *in* and so $e(x)$ is *out*.

Case 3

This is the case of $a = out$ and $c = in$. It is the mirror image of Case 2 and follows from the symmetry of the Figure with a and c swapped.

Case 4

This is where $a = in$ and $c = in$. This case should come out inconsistent. Indeed, in this case both \bar{a}, \bar{b} are *out*.

Hence since $\bar{x} = out$ we get $e(x, a)$ and $e(x, c)$ are *in* and since they attack c and a respectively, c and a cannot be *in*. An inconsistency. We also have that \bar{a}, \bar{c} are *in* and $d(x)$ is *out*.

We now need to check what happens when Figure 50 is embedded inside a larger network. Figure 50 arose from the attempt to eliminate the disjunctive attack of Figure 46. So if Figure 46 is part of a larger network then Figure 50 will also be in the larger network. However, the larger network interacts (attacks or is being attacked) only by $\{x, a, c\}$ and not the new points. So it is crucial for us to show that if the labels of $\{x, a, c\}$ are fixed by the larger

network then the labels of the new points are unique and they are consistent with the labels of $\{x, a, c\}$.

LEMMA 3.2. *In Figure 50, if we fix the value $\{\text{in}, ?, \text{out}\}$ of $\{a, c, x\}$ then the value of the other points $\{\bar{a}, \bar{c}, \bar{x}, \bar{a}, \bar{c}, e(x), e(x, a), e(x, c)\}$ are also uniquely fixed, and are consistent with the values of $\{a, c, x\}$.*

PROOF. We need to show that if the values of $\{a, c, x\}$ are fixed then the values of the new points are uniquely determined.

Let \mathbf{N} be a general network in which Figure 46 is a subnetwork. This means that there may be attacks on $\{a, c, x\}$ from other nodes of the network \mathbf{N} , say from $\{d_1, \dots, d_k\}$ and also that there may be attacks emanating from $\{a, c, x\}$ individually or jointly with others onto nodes $\{d'_1, \dots, d'_{k'}\}$ in \mathbf{N} .

We are now replacing in \mathbf{N} Figure 46 by Figure 50. The word ‘replacing’ is not accurate. Figure 46 (i.e. the subnetwork of \mathbf{N} comprising of nodes $\{a, c, x\}$ and the connections between them) remains in \mathbf{N} , we are adding new points and connections to Figure 46 to form Figure 50 a part of \mathbf{N} .

We want to show that any acceptable labelling λ on \mathbf{N} which gives values to $\{a, c, x\}$ can be uniquely extended to the new points of Figure 50 in a manner consistent with λ . Call the new extension λ^* .

We need to make a case analysis on the values of $\lambda(a), \lambda(c), \lambda(x)$.

We have to check all cases. We follow the cases in three groups $x = 1, x = 0$ and $x = ?$. For each group we check all cases of a and c as in Table 1.

Group $x = \text{in}$

Table 1 suggests 9 cases for a and c . However not all are possible in \mathbf{N} . Since λ is an acceptable labelling and $x = \text{in}$, only cases in which either $a = \text{out}$ or $c = \text{out}$ are possible.

Thus we need examine only cases 1, 2, 3, 5, 8. Cases 1, 2, 3 have already been examined in Example 3.1. They are OK and are internally consistent. Cases 5 and 8 are completely symmetrical (since Figure 50 is symmetrical in a and c). So we need examine only Case 5.

Case 5

$a = \text{out}, c = ?$

First we see which values for the new points are forced by this case.

$$\begin{aligned} a = \text{out} &\Rightarrow \bar{a} = \text{in} \\ \bar{a} = \text{in} &\Rightarrow \bar{\bar{a}} = \text{out} \\ \bar{a} = \text{in} &\Rightarrow e(x, a) = \text{out}. \end{aligned}$$

Claim

$e(x)$ cannot be *in*!

Table 1.

case	c	a
1	out	out
2	out	in
3	in	out
4	in	in
5	?	out
6	?	in
7	in	?
8	out	?
9	?	?

Otherwise, since $\bar{a} = out$ then if $e(x)$ were *in* then $\bar{c} = out$, hence $\bar{c} = in$ hence c is *out*. But we are given that $c = ?$. So $e(x)$ is *not in*.

Therefore, since $e(x, a) = out$ and $e(x)$ is either *out* or $?$ we have a situation which is consistent with $c = ?$. We now use the fact that $c = ?$ to get $\bar{c} = ?$ and hence $\bar{c} = ?$, and hence $e(x) = ?$. This establishes a unique value for $e(x)$. We continue, since $\bar{c} = out$ and $\bar{c} = ?$, we get $e(x, c) = ?$.

We have that $e(x, c)$ attacks a . Does this contradict $a = out$? The answer is no. a is *out* because of some attack from \mathbf{N} . So we got for Case 5 unique consistent values for the new points.

Group $x = out$

Since $x = out$, there are no attacks from x on a and c , therefore all values of $\{a, c\}$ of Table 1 are possible. We have to examine all cases.

We begin by checking what values are forced on the new points by the fact that $x = out$. We get

$$\begin{aligned} x = out &\Rightarrow \bar{x} = in \\ \bar{x} = in &\Rightarrow e(x, a) = e(x, c) = e(x) = out \end{aligned}$$

Since there are no attacks from the new points of Figure 50 on the points $\{a, c\}$, their value consistently stand in the expanded network.

If $\lambda = 1, 0, ?$ use the notation $1 - \lambda$ to mean $-, 1, ?$ respectively. Thus the value of \bar{a} is determined as $1 - \lambda(a)$ and the value of \bar{c} is determined as $1 - \lambda(c)$. Thus the value of \bar{a} is the same as the value of a and the value of \bar{c} is the same as the value of c .

Hence for all cases of Table 1 we see that the values of the new points are unique and are consistent with the values of $\{a, c\}$ for our group case of $x = out$.

Group $x = ?$

If $x = ?$ and x disjunctively attacks a and c , we cannot have the case of $a = c = in$, because maybe x can be in . Similarly we cannot have the cases of $a = in, c = ?$ nor $a = ?$ and $c = in$.

So let us see first what we can get which holds for all cases

$$x = ? \Rightarrow \bar{x} = ?$$

This is all we can get.

We have to check cases 1–3, 5, 8, 9.

Case 1

$c = out$ and $a = out$.

$$\begin{aligned} a = out &\Rightarrow \bar{a} = in \text{ and } e(x, a) = out \\ c = out &\Rightarrow \bar{c} = in \text{ and } e(x, c) = out \\ \bar{a} = in &\Rightarrow \bar{\bar{a}} = out \\ \bar{c} = in &\Rightarrow \bar{\bar{c}} = out \\ \bar{x} = ? \text{ and } \bar{\bar{a}} = \bar{\bar{c}} = out &\Rightarrow e(x) = ? \end{aligned}$$

Now a and c are out, they are attacked by $e(x) = ?$. This is still consistent.

So this case is OK.

Case 2

$c = out$ and $a = in$.

$$\begin{aligned} \bar{\bar{c}} = out &\Rightarrow \bar{c} = in \\ \bar{c} = in &\Rightarrow \bar{\bar{c}} = out \\ \bar{c} = in &\rightarrow e(x, c) = out \\ a = in &\Rightarrow \bar{a} = out \\ \bar{a} = out &\Rightarrow \bar{\bar{a}} = in \text{ and } e(x, a) = out \\ \bar{\bar{a}} = in &\Rightarrow e(x) = out. \end{aligned}$$

We now get unique values for the new points. We have consistency because there are no attacks on $\{a, c\}$ from the new points.

Case 3

$c = in$ and $a = out$.

This is the symmetrical case of Case 2.

Case 5

$c = ?$ and $a = out$.

We have $x = ?$ and $\bar{x} = ?$

$$\begin{aligned}
 a &= out \Rightarrow \bar{a} = in \\
 \bar{a} &= in \Rightarrow e(x, a) = out \\
 c &=? \Rightarrow \bar{c}=? \\
 \bar{x} &=? \text{ and } \bar{c}=? \Rightarrow e(x, c) = ? \\
 \bar{a} &= in \Rightarrow \bar{\bar{a}} = out \\
 \bar{c} &=? \Rightarrow \bar{\bar{c}}=? \\
 \bar{\bar{a}} &= out \text{ and } \bar{\bar{c}}=? \Rightarrow e(x) = ?
 \end{aligned}$$

We now got unique values for all new points. Do we have consistency?

$a = out$ is attacked by

$$e(x) = e(x, c) = ?$$

Hence a remains *out*. We have consistency here.

$c = ?$ is attacked by $e(x, a) = out$, so we maintain consistency.

Case 8

$c = out$ and $a = ?$

This case is symmetrical with Case 5, with a and c interchanged. By symmetry it is OK.

Case 9

$a = ?$ and $c = ?$

In this case we get everything has value ? and we have no problem. ■

The situation of moving from Figure 46 to Figure 50 is typical and the fact that we dealt with an attack from x on $\{a, c\}$ and we did not deal with attacks from a general set $X = \{x_1, \dots, x_n\}$ on a general set $Y = \{y_1, \dots, y_m\}$ does not incur any loss of generality. In the general case we proceed similarly. We need to add the auxiliary points $\bar{\bar{y}}_1(X), \dots, \bar{\bar{y}}_m(X)$ add attacks of y_i on $\bar{\bar{y}}_i(X)$ and add joint attacks as in Figure 51.

3.2. Eliminating disjunctive and joint attacks

We are now ready, following our conceptual discussion in Section 3.1, to give a series of definitions and Lemmas showing how networks with joint and disjunctive attacks can be reduced to ordinary Dung networks with only point-to-point attacks. The reduction is done with the help of auxiliary points, using the intuition described in Figure 52.

DEFINITION 3.3 (Critical subsets). Let $\mathbf{P}_i = (S_i, R_i)$ be two networks. Suppose all points of network \mathbf{P}_2 are embedded inside network \mathbf{P}_1 . So S_2 is

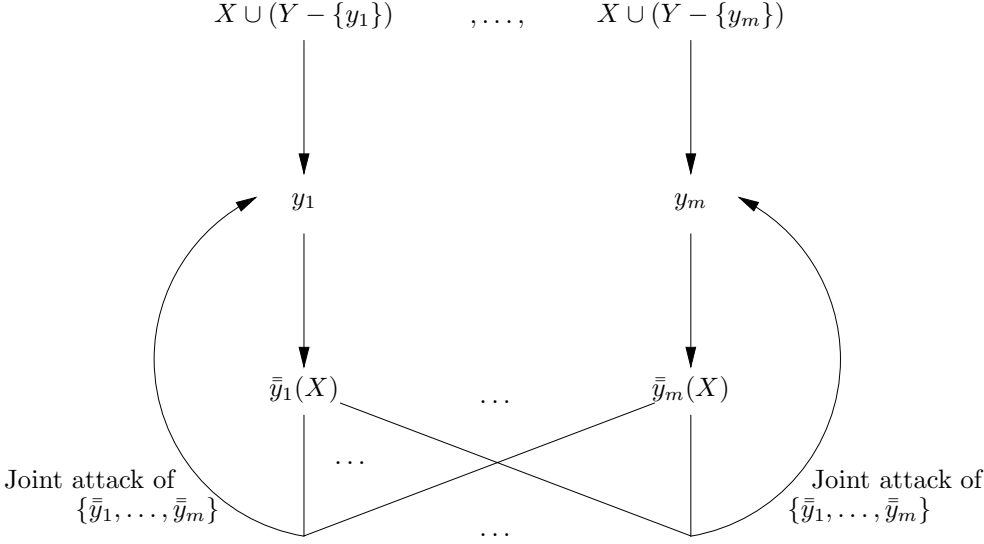


Figure 51.

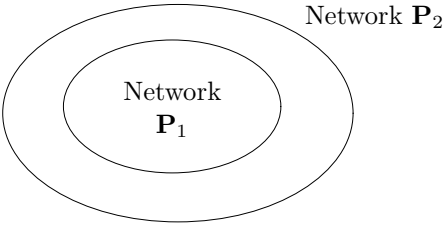


Figure 52.

a subset of S_1 . We say that S_2 is a *critical* subset of S_1 iff every Caminada labelling on S_2 can be extended *uniquely* to a labelling on S_1 . This means that the additional nodes of S_1 only help clarify what is going on in S_2 and do not add any additional information. Any Caminada labellings of S_1 which agree on S_2 must be equal.

This sort of characterisation is known from model theory. A set of classical models \mathcal{M}_2 in a language \mathbb{L}_2 is said to be EC_Δ iff it can be characterised as the set of all models of a first-order theory Δ_2 of \mathbb{L}_2 . The set of models is said to be PC_Δ iff we can extend the language \mathbb{L}_2 such that the set there exists a theory Δ_1 in the language \mathbb{L}_1 such that any model \mathbf{m}_2 of \mathcal{M}_2 can be obtained as the restriction $\mathbf{m}_2 = \mathbf{m}_1 \upharpoonright \mathbb{L}_2$ to \mathbb{L}_2 of a unique model \mathbf{m}_1 of Δ_1 . Furthermore, all restrictions to \mathbb{L}_2 of models of Δ_1 are models of Δ_2 .

We are now ready to push on with our reduction.

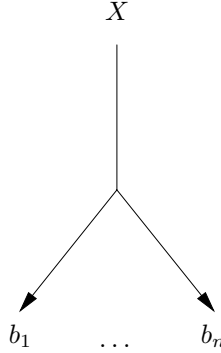


Figure 53.

DEFINITION 3.4 (Eliminating disjunctive attacks). Let $\mathbf{P} = (S, S^0, R)$. Let $S_1 = S \cup \{\bar{s}(X) | s \in S, X \subseteq S, X \neq \emptyset\}$.

Let $\mathbf{P}_1 = (S_1, S_1^0, R_1)$ be defined as follows, see Figure 51:

1. Let XR_1a if XRa holds.
2. If XRy holds then let $X \cup (Y - \{y\})R_1y$ hold for every $y \in Y$.
3. Let $sR_1\bar{s}(X)$ hold for every $s \in S, X \subseteq S, X \neq \emptyset$
4. If XRy holds let $\{\bar{y}(X) | y \in Y\}R_1y$ hold of each $y \in Y$.

LEMMA 3.5. For any $\lambda : S \mapsto \{0, 1, ?\}$ we have that λ is proper labelling on \mathbf{P} iff λ is proper on \mathbf{P}_1 .

PROOF. One can give a direct proof of this theorem by working directly of Figure 51, converting it to the analog of Figure 50. However a simpler rout is to use Lemma 3.2 for the case of disjunctive attacks involving only two elements. All we need to do is to convert a disjunctive attack on $n \geq 3$ elements, as in Figure 53 to a disjunctive attack on $n - 1$ elements as in Figure 54. We use auxiliary points as indicated.

We can therefore assume that we are dealing only with disjunctive attacks on two points, since all other disjunctive attacks can be reduced to two points by repeated applications of the above procedure.

The case of disjunctive attacks of two points was dealt with by Figure 50 and Lemma 3.2. ■

DEFINITION 3.6 (Eliminating joint attacks). Let $\mathbf{P} = (S, S^0, R)$ be a joint attack network. Define an ordinary network $\mathbf{P}^* = (S^*, R^*)$, with $S \subseteq S^*$ and S^*, R^* as follows. We add to S the following additional groups of points.

Group G_1

For every $s \in S$ a new node \bar{s}

Group G_2

For every $X \subseteq S, X$ finite with two points or more add the node $e(X)$.

Let $S^* = S \cup G_1 \cup G_2$.

Define R^* on S^* as follows.

Assume $\{a_1, \dots, a_n\}Rb_j$ hold for $n \geq 2, j = 1, \dots, k$, where b_1, \dots, b_k are all the nodes jointly attacked by $\{a_1, \dots, a_n\}$. Figure 55 shows the situation.

Note that we cannot represent this situation by writing Figure 56 because that would be a disjunctive attack.

We define R^* according to Figure 57. Thus Figure 55 with R becomes Figure 57 with R^* .

We have that the following hold in R^* :

1. $sR^*\bar{s}$ for any $s \in S$.
2. tR^*s whenever $\{t\}Rs$
3. If $\{a_1, \dots, a_n\}Rb_j$, holds with $n \geq 2$, and $j = 1, \dots, m$ then let $\bar{a}_iR^*e(a_1, \dots, a_n)$ hold for $i = 1, \dots, n$ and let $e(a_1, \dots, a_n)R^*b_j$ hold, for $j = 1, \dots, m$.

LEMMA 3.7. Let \mathbf{P} and \mathbf{P}^* be as in Definition 3.6. Observe the following

1. Since s is the only attacker of \bar{s} , we have for any Caminada labelling that

$$\begin{aligned} \lambda(s) = 1 & \text{ iff } \lambda(\bar{s}) = 0 \\ \lambda(x) = ? & \text{ iff } \lambda(\bar{s}) = ? \end{aligned}$$

2. In Figure 57

$$\begin{aligned} \lambda(e(a_1, \dots, a_n)) &= 1 \text{ iff} \\ \lambda(a_1) = \lambda(a_2) = \dots = \lambda(a_n) &= 1 \end{aligned}$$

3. If $\lambda(a_i) = 1$ for $i = 1, \dots, n$, then $\lambda(b_j) = 0, j = 1, \dots, m$.
4. If for some $i, \lambda(a_i) = 0$ then $\lambda(e(a_1, \dots, a_n)) = 0$ and then there is no attack from $e(a_1, \dots, a_m)$ on any $b_j, j = 1, \dots, m$.
5. Any Caminada–Gabbay labelling function on λ^* on S^* induces a labelling $\lambda = \lambda^* \upharpoonright S$ on S .

To explain the relationship between \mathbf{P} and \mathbf{P}^* we need some general concepts.

DEFINITION 3.8. Let $\mathbf{P} = (S, R)$ be ordinary Dung argumentation networks. Let $E \subseteq S$ be a set of nodes. We say that E is a *critical* set in \mathbf{P} iff the following holds:

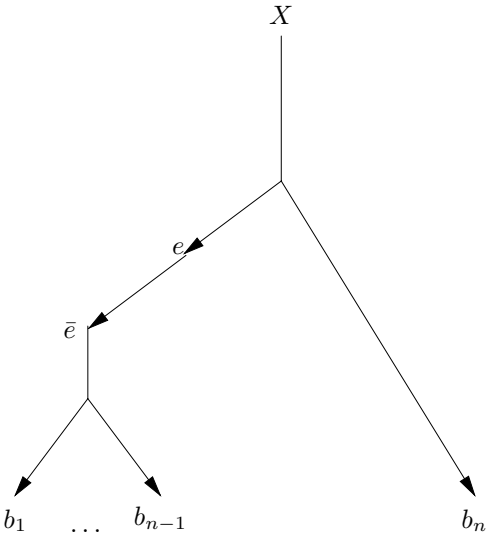


Figure 54.

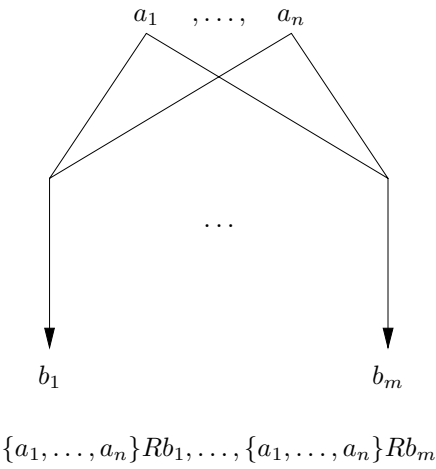


Figure 55.

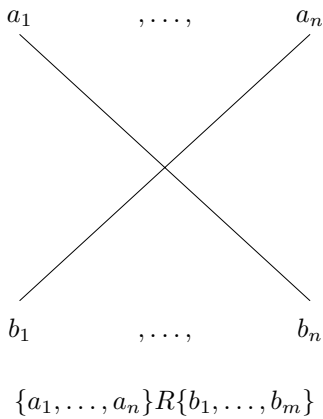


Figure 56.

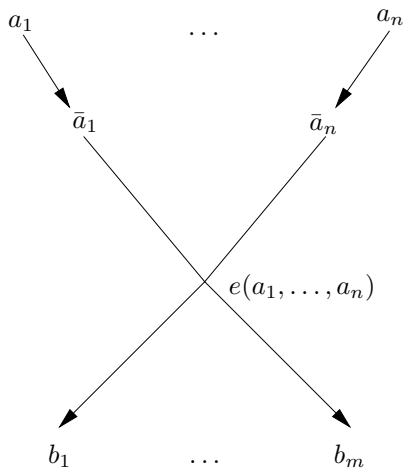


Figure 57.

- (*) For any two Caminada labelling λ_1 and λ_2 on \mathbf{P} , if λ_1 and λ_2 agree on E then $\lambda_1 = \lambda_2$

LEMMA 3.9. *In Definition 3.6, S is critical in (S^*, R^*) .*

PROOF. Follows from Lemma 3.7. ■

REMARK 3.10 (Labelling of higher order networks). *To show the existence of labelling for higher order networks $\mathbf{P} = (S, S^0, R)$, we reduce the network to an ordinary network $\mathbf{P}^* = (S^*, R^*)$ by eliminating first the disjunctive attacks and then the joint attacks.*

From our sequence of Lemmas we know that $S \subseteq S^$ is critical in P^* . Therefore any labelling λ^* on S^* induces a labelling $\lambda = \lambda^* \upharpoonright S$ on S which is acceptable in \mathbf{P} . Furthermore, λ^* can be uniquely retrieved from λ and thus any λ on S can be expanded to a unique λ^* on S^* .*

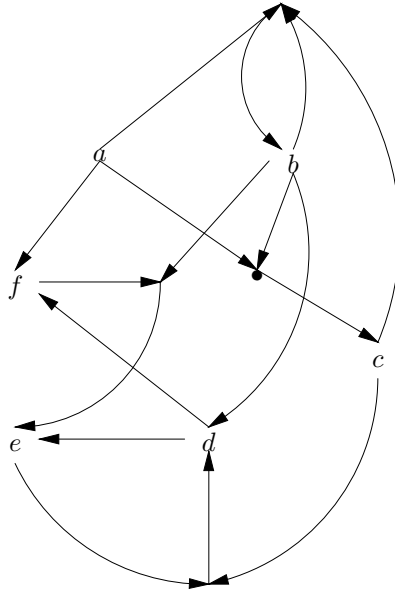


Figure 58.

EXAMPLE 3.11. Figure 58 displays the following network from [19].

$$S = \{a, b, c, d, e, f\}$$

R is defined as follows

$$\begin{aligned} &\{a, c, d\}Rb \\ &\{ab\}Rc \\ &\{b\}Rd \\ &\{c, e\}Rd \\ &\{d\}Re \\ &\{b, f\}Re \\ &\{a\}Rf \\ &\{d\}Rf \end{aligned}$$

The grounded extension is $\{a\}$. The preferred extensions are $\{a, b, e\}$ and $\{a, c, d\}$.

We now construct our reduct ordinary network. We add points $\bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{e}, \bar{f}$ and $e(a, b, d), e(a, b), e(c, e)$ and $e(b, f)$ and get Figure 59.

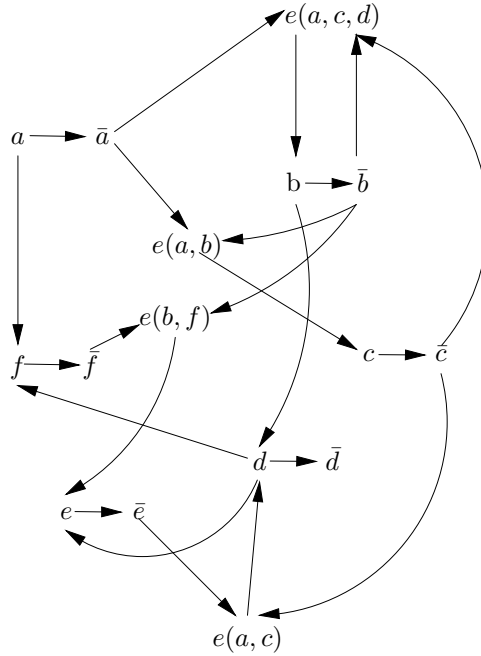


Figure 59.

We have the following attacks:

$$\begin{aligned}
 a &\rightarrow \bar{a} \\
 b &\rightarrow \bar{b} \\
 c &\rightarrow \bar{c} \\
 d &\rightarrow \bar{d} \\
 e &\rightarrow \bar{e} \\
 f &\rightarrow \bar{f} \\
 \bar{a} &\rightarrow e(a, c, d) \\
 \bar{c} &\rightarrow e(a, c, d) \\
 \bar{d} &\rightarrow e(a, c, d) \\
 e(a, c, d) &\rightarrow b \\
 \bar{a} &\rightarrow e(a, b) \\
 \bar{b} &\rightarrow e(a, b) \\
 e(a, b) &\rightarrow c \\
 b &\rightarrow d \\
 \bar{c} &\rightarrow e(c, e) \\
 \bar{e} &\rightarrow e(c, e) \\
 e(c, e) &\rightarrow d \\
 d &\rightarrow e \\
 \bar{b} &\rightarrow e(b, f) \\
 \bar{f} &\rightarrow e(b, f) \\
 e(b, f) &\rightarrow e \\
 a &\rightarrow f \\
 d &\rightarrow f
 \end{aligned}$$

Graphically it is very easy to convert any joint network to an ordinary network as follows.

1. First replace any node x by $x \rightarrow \bar{x}$ and let any arrows emanating from x to emanate now from \bar{x} . Any arrows targeting x remain as is.
2. Any joint attack from a_1, \dots, a_n on b have arrows emanating from each a_i (now emanating from \bar{a}_i) to an unnamed junction where they all meet and then a single arrow go to b as in Figure 60. Name the junction $e(a_1, \dots, a_n)$ and let the arrows be now attacking arrows of an ordinary network. We get Figure 61.

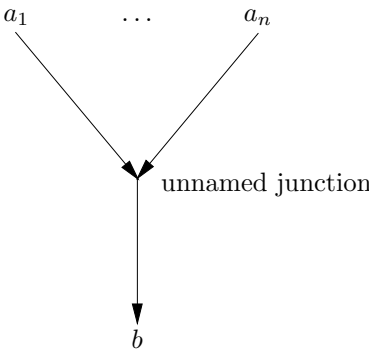


Figure 60.

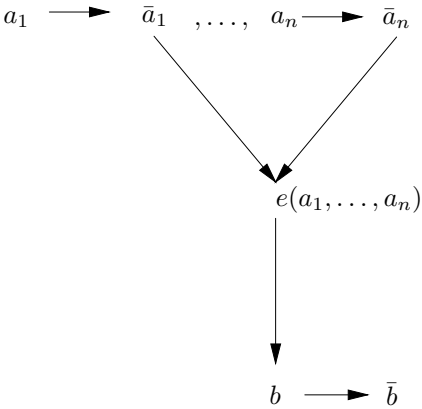


Figure 61.

4. Comparison and discussion

4.1. Comparison with Nielsen and Parsons

We compare this work with the paper of Nielsen and Parsons [20, 19]. Nielsen and Parsons put forward a system where joint attacks are possible. They introduce attacks of the form XRy where X is a non-empty set of arguments and y is an argument.

The attack XRY , where Y is a set of arguments is reduced to

$$XRY \text{ iff } XRy \text{ for some } y \in Y.$$

This is the same as our approach for the case in which the networks we are dealing with have no arrows in them and in which there are no disjunctive attacks. The point of view of Nielsen and Parsons arises from a qualitative argument in favour of joint attacks. They argue their case convincingly and proceed to develop the theory beautifully in their first paper [20]. In the second paper [19] they continue with an algorithm for computing extensions.

We arrived at these attacks from a different point of view. We are substituting one network in another. We thus get a network attacking another network. Since the networks may have points in common they can internally influence one another.

To obtain the Nielsen and Parsons case our networks should have no internal arrows, and no disjunctive attacks. So we interpret an attack on a network as in Figure 46 reading (r2). Another difference between the two approaches is the reduction we make of the joint attack networks to ordinary Dung networks. The original network becomes a critical subset in an ordinary Dung network and so we can compute the extensions of our original network by computing the extensions of the target network.

We would like to quote and criticise a statement of Nielsen–Parsons in their paper [20] they say:

We claim that it is never necessary to specify a non-singleton set of arguments as attacked, as in $\{A_1, \dots, A_n\} \triangleright \{B_1, \dots, B_m\}$: If collective defeat is taken to heart, the attack can be reformulates as a series of attacks

$$\begin{array}{c} \{A_1, \dots, A_n\} \triangleright B_1 \\ \vdots \\ \{A_1, \dots, A_n\} \triangleright B_m \end{array}$$

It is easily seen that the above attacks would imply the attack, which is intended, as the validity of the A -arguments would ensure that none of the B -arguments are valid.

If instead indeterministic defeat is required, the attack can be reformulated as

$$\{A_1, \dots, B_n, B_2, \dots, B_m, \} \triangleright B_1,$$

which ensures that in case the A arguments are valid, then B_1 cannot be a valid argument if the remaining B -arguments are also true, thus preventing the entire set of B -arguments from being valid at once, if the A -arguments are true.

Nielsen and Parsons use reading (r3) of Figure 46 (to use our terminology), namely they use Figure 48. Their notion of ‘collective defeat’ requires reading (r2) and not (r3) as Nielsen and Parsons claim.

4.2. Connection with attacks on attacks

In our paper [1] we introduced the notion of attack on attack. Figure 62 is an example.

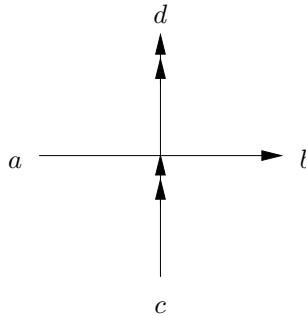


Figure 62.

The network contains the arguments $\{a, b, c, d\}$. a attacks b . Denote this attack by ‘ $a \rightarrow b$ ’. The attack ‘ $a \rightarrow b$ ’ is being attacked by c and it itself attacks the node d .

Our question is how does this relate to our joint and disjunctive attack networks?

Consider Figures 63 and 64.

Can we view Figure 62 as the result of substituting Figure 64 for x in Figure 63? The answer is that if we do that we get the wrong meaning. The subject is studied extensively in [1] and [13]. The actual meaning of Figure 62 is that c attacks the argument (denoted by ‘ $a \rightarrow b$ ’) motivating and describing why a attacks b , and this very argument actually attacks d .

So the correct interpretation of Figure 64 is Figure 65.

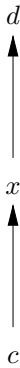


Figure 63.

$$a \longrightarrow b$$

Figure 64.

a and ' $a \rightarrow b$ ' mount a joint attack on b . So Figure 62 can be presented as Figure 66.

Of course if we want to be very pedantic we can say that c , which attacks the ' $a \rightarrow b$ ' component of this joint attack, is attacking ' $a \rightarrow b$ ' jointly with ' $(c \rightarrow (a \rightarrow b))$ ' argument. Also ' $a \rightarrow b$ ' which is attacking d is attacking it jointly with " $((a \rightarrow b) \rightarrow d)$ " argument. So the correct figure corresponding to Figure 62 is Figure 67.

However, since the attacks of c on ' $a \rightarrow b$ ' and of ' $a \rightarrow b$ ' on d are not themselves being attacked, we do not need to unfold them in Figure 67 and Figure 66 is sufficient.

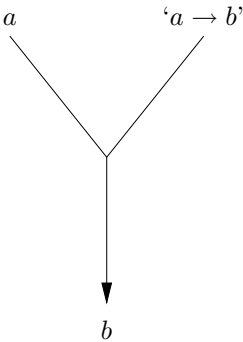


Figure 65.

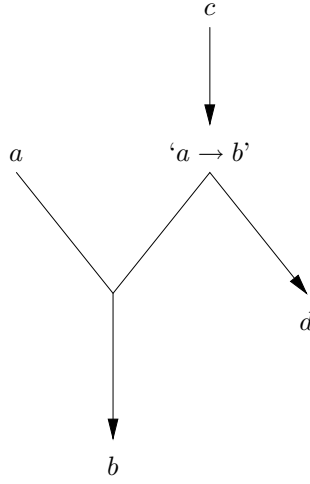


Figure 66.

Note that if we systematically rewrite attacks on attacks on attacks, etc., as joint attacks and then use the semantics of Section 3 to define extensions, we end up with semantics for all these higher level attacks. This is done directly in [13].

4.3. Flow argumentation networks

This subsection deals with Figure 41. To explain Figure 41, we need a consistent point of view to explain the case of disjunctive attacks as in Figure 46. For Figure 46 we know that we require that

(*) $x = in$ implies $a = out$ or $c = out$

The questions relevant to our understanding of Figure 41 are the following

Question 1

Does x attack each of a and c and expects at least one of them to succeed (reading (r2) of Figure 41 in Section 3.1) or does x attack only the set $\{a, c\}$ and expects (*) above to hold? (Reading (r1) of Figure 41 in Section 3.1).

Obviously to be able to give Figure 41 any meaning we must adopt the view that x attacks each of a and c (reading (r2)).

We now ask our second question.

Question 2

When the attack of x on one of $\{a, c\}$ succeeds, e.g. $a = out$, is x still attacking c ? In other words x is still attacking even though x does not care

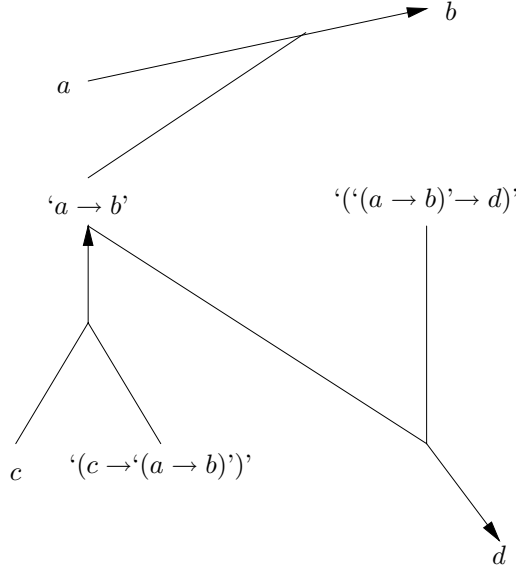


Figure 67.

if the attack succeeds? If the answer is no, and there is no attack on c , then this means that in Figure 41, the labelling

1. $x = in, b = in, a = out, c = in$

is acceptable, because the success of the attack on a ($a = out$) entails that there is no attack from x onto c and hence the joint attack with b on c fails because x is not attacking and hence $c = in$.² If the answer is yes, and indeed there is a attack on c then (1) is not acceptable and (2) is acceptable.

2. $x = in, b = in, a = out$ and $c = out$

Of course, both views will accept (3) below.

3. $x = in, b = in, a = in$ and $c = out$

We adopt the view that x sends attacks in the direction of a and c and expects at least one of them to succeed (this is the (r2) reading of Figure 46 in Section 3.1).³

²In this case, $a = out$ makes x attack (with b) on c a voluntary attack. We investigate voluntary attacks in [4].

³Imagine mother wants to buy a special toy for her little girl for Christmas. There is great demand for the toy which is very popular and people queue up early in the morning

Notice the words we use. We talk about ‘ x sending attacks in the direction of y ’. We speak of ‘joint attacks’ and ‘disjunctive attacks’. We need to formalise our intuitive model. We found that the best way to represent what is going on is in terms of ‘flow’, the attacks flow along the edges of the network. So we call these ‘flow argumentation networks’.

It would help if we consider an example. Consider Figure 68, which is an expansion of Figure 41, with a new twist to it. The arguments of Figure 68 are $\{x, a, b, c, d, e\}$ and we also gave names to the flow (flow of attacks) junctions $\{\alpha, \beta, \gamma, \delta\}$. These are not arguments, just input output nodes in the graph, to help us follow the attacks flows.

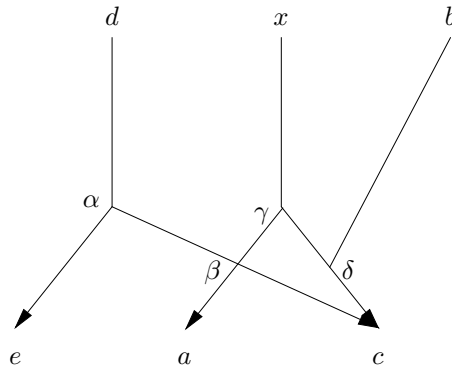


Figure 68.

In Figure 68, d sends a disjunctive attack through junction α . The attack splits into an attack on e and an attack in the direction of a and c . x at the same time sends a disjunctive attack through junction γ which splits one in the direction of a and one in the direction of c . The x attack going to c joins forces with the attack on c emanating from b . At junction δ they join and become a joint attack on c .

The attack of x in the direction of a meets the attack of d in the direction of a and c at junction β and decide to mount a joint attack emanating from β on a and c . The flow from γ to δ is an attack of x intended for a ,

at toy shops to make sure that they can get the product. So mother sends her husband and her brother to two different shops to queue up early in the morning in the hope that at least one of them will get to buy the toy before stocks run out for the day. Mother would be happy if at least one of them succeeds. Under this view each one of the two men is there buying a toy.

The other option is for mother to ask them to call the other mobile phone the minute one of them secures the toy. In that case the other may not buy the toy. This is the other option.

at β it joined the flow $d \rightarrow \alpha \rightarrow \beta$ and became a joint attack from β on a and on c . We can say x did not mind forming this coalition with d and adding c to the attack because x is attacking c anyway through the direction $x \rightarrow \gamma \rightarrow \delta \rightarrow c$.

The above description is in literary prose, giving intentions to the arrows in terms of flows. How do we do this formally? And how do we calculate labelling?

Suppose we label $c = out$, $a = in$, $e = in$. Is this OK with $d = x = b = in$? Or maybe we need also $e = out$ not $e = in$?

We need a formal model which follows the flow using $\alpha, \beta, \gamma, \delta$.

We leave this as the subject of another paper.

References

- [1] BARRINGER, H., D. M. GABBAY, and J. WOODS, ‘Temporal dynamics of argumentation networks’, in D. Hutter and W. Stephan (eds.), *Mechanising Mathematical Reasoning*, LNCS 2605, Springer, 2005, pp. 58–98.
- [2] BARRINGER, H., D. M. GABBAY, and J. WOODS, ‘Network modalities’, in G. Gross and K. Schulz (eds.), *Linguistics, Computer Science and Language Processing: Festschrift for Franz Guenther on the Occasion of His 60th Birthday*, College Publications, 2008, pp. 79–102.
- [3] BOELLA, G., D. M. GABBAY, L. VAN DER TORRE and S. VILLATA, ‘Meta-argumentation part 1’, *Studia Logica*, 93(2-3): 297–355, 2009, this issue.
- [4] BOELLA, G., D. M. GABBAY, L. VAN DER TORRE, and S. VILLATA, ‘An argumentation based approach to coalition formation using voluntary attacks’, In preparation.
- [5] CAMINADA, M., and D. M. GABBAY, ‘A logical account of formal argumentation’, *Studia Logica*, 93(2-3): 109–145, 2009, this issue.
- [6] CARNIELLI, W., M. CONIGLIO, D. M. GABBAY, P. GOUVEIA, and C. SERNADAS, *Analysis and Synthesis of Logics*, 500 pp., Springer, 2007.
- [7] DUNG, P. M., ‘On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n -person games’, *Artificial Intelligence* 77: 321–357, 1995.
- [8] ENGESSER, K., D. M. GABBAY, and D. LEHMANN, *A New Approach to Quantum Logic*, 300 pp., College Publications, 2008.
- [9] GABBAY, D. M., *Labelled Deduction Systems*, Oxford University Press, 1996.
- [10] GABBAY, D. M., *Logic for AI and Information Technology*, 500 pp., College Publications, October 2007.
- [11] GABBAY, D. M., ‘Reactive Kripke models and contrary to duty norms’, in R. van der Meyden and L. van der Torre (eds.), *Proceedings of DEON 2008*, Deontic Logic in Computer Science, LNAI 5076, Springer, 2008, pp. 155–173.
- [12] GABBAY, D. M., ‘Modal provability foundations for argumentation networks’, *Studia Logica*, 93(2-3): 181–198, 2009, this issue.
- [13] GABBAY, D. M., ‘Semantics for higher level attacks in extended argumentation frames, Part 1: Overview’, *Studia Logica*, 93(2-3): 357–381, 2009, this issue.

- [14] GABBAY, D. M., *The Leverhulme Lectures in Logic*, 200 pp. Draft available.
- [15] GABBAY, D. M., and A. GARCEZ, ‘Logical modes of attack in argumentation networks’, *Studia Logica*, 93(2-3): 199–230, 2009, this issue.
- [16] GABBAY, D. M., R. SCHMIDT and A. SZALAS, *Second-Order quantifier Elimination*, 400 pp., College Publications, 2008.
- [17] GABBAY, D. M., and A. SZALAS, ‘Second-order Quantifier Elimination in higher-order contexts with applications to the semantical analysis of conditionals’, *Studia Logica* 87: 37–50, 2007.
- [18] GABBAY, D. M., and A. SZALAS, ‘Annotation theories over finite graphs’, *Studia Logica*, 93(2-3): 147–180, 2009, this issue.
- [19] NIELSEN, S. H., and S. PARSONS, ‘Computing preferred extensions for argumentation systems with sets of attacking arguments’, in *Proceedings of the 2006 Conference on Computational Models of Argument: Proceedings of COMMA 2007*, 2006, pp. 97–108.
- [20] NIELSEN, S. H., and S. PARSONS, *A Generalization of Dung’s Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments*, LNCS Vol. 4766, Springer, 2007, pp. 54–73.
- [21] WU, Y., M. CAMINADA, and D. GABBAY, ‘Complete extensions in argumentation coincide with 3-valued stable models in logic programming’, *Studia Logica*, 93(2-3): 383–403, 2009, this issue.

DOV M. GABBAY
King’s College, London
Bar-Alan University, Israel
University of Luxembourg
`dov.gabbay@kcl.ac.uk`