

Modal Provability Foundations for Argumentation Networks

Paper 331-h

D. M. Gabbay

First draft: October 2008
Second draft: February 2009

Abstract

Given an argumentation network we associate with it a modal formula representing the ‘logical content’ of the network. We show a one-to-one correspondence between all possible complete Caminada labellings of the network and all possible models of the formula.

1 Introduction

Let \mathbf{P} be an argumentation network. Figures 1, 2, 3 and 4 are typical examples.

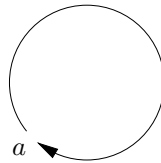


Figure 1:

The network \mathbf{P} has logical contents, saying which arguments are winning (denoted by a is ‘on’ or $a = 1$), which are losing (denoted by a is ‘off’ or $a = 0$) and which are undecided (denoted by a is ‘undecided’ or $a = ?$).

The content (or extensions) can be obtained from a Caminada labelling.

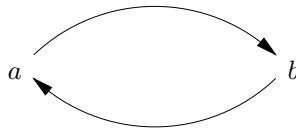


Figure 2:

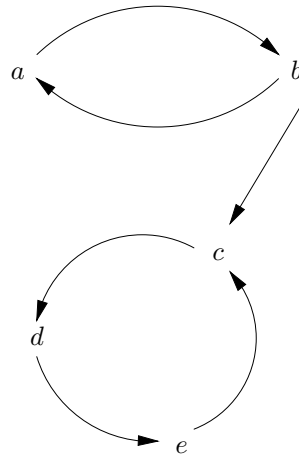


Figure 3:

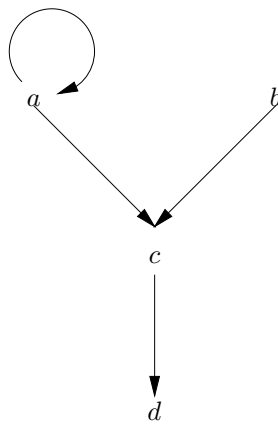


Figure 4:

Definition 1.1 (Caminada labelling) A function $\mathbf{f} : \mathbf{P} \mapsto \{1, 0, ?\}$ is a Caminada labelling iff the following holds.

1. If x is an initial point (no arrow leads into x) then $\mathbf{f}(x) = 1$.
2. If y_1, \dots, y_n are all nodes with arrows leading into y then
 - 2.1. if for some i , $\mathbf{f}(y_i) = 1$ then $\mathbf{f}(y) = 0$
 - 2.2. if for all i , $\mathbf{f}(y_i) = 0$ then $\mathbf{f}(y) = 1$
 - 2.3. if for all i , $\mathbf{f}(y_i)$ is either 0 or ? and for at least one of y_i $\mathbf{f}(y_i) = ?$ then $\mathbf{f}(y) = ?$.

The extension obtained from a Caminada labelling is $\{x | \mathbf{f}(x) = 1\}$. The family of extensions for any particular algorithm is obtained from conditions on the Caminada labelling. For example, let $E_{\mathbf{f}}^y = \{x | \mathbf{f}(x) = y\}$, where $y = 0, 1, ?$. Order $\{E_{\mathbf{f}}^y | \mathbf{f} \text{ a Caminada function}\}$, for y fixed, by inclusion. We have the following:

Grounded extension: $E_{\mathbf{f}}^1$ and $E_{\mathbf{f}}^-$ are minimal. $E_{\mathbf{f}}^?$ is maximal.

Preferred extension: $E_{\mathbf{f}}^1$ and $E_{\mathbf{f}}^0$ are maximal.

Complete extension: Same as complete labelling.

Stable extension: $E_{\mathbf{f}}^? = \emptyset$, i.e. no undecided.

Semi-stable extension: $E_{\mathbf{f}}^?$ is minimal.

See [5] for a survey.

This kind of content depends on an algorithm yielding extensions and there are several types of algorithms, giving several types of extensions. Different networks may admit some extensions and possibly not others.

For example

1. Network \mathbf{P}_1 of Figure 1 has no extensions.
2. The network \mathbf{P}_2 of Figure 2 has the following types of extensions:
 - Grounded: \emptyset
 - Preferred: $\{a\}, \{b\}$
 - Complete: $\emptyset, \{a\}, \{b\}$.
3. The network \mathbf{P}_3 of Figure 3 has the following types of extensions:
 - Grounded: \emptyset
 - Preferred: $\{a\}, \{b, d\}$
 - Complete: $\emptyset, \{b, d\}$.
4. The network \mathbf{P}_4 in Figure 4 has the following types of extensions:
 - All types except stable: $\{b, d\}$



Figure 5:

| | Type 1 | Type 0 | Type ? |
|---|--------|--------|--------|
| 3 | ⊤ | ⊥ | ⊤ |
| ↑ | ↑ | ↑ | ↑ |
| 2 | ⊤ | ⊥ | ⊥ |
| ↑ | ↑ | ↑ | ↑ |
| 1 | ⊤ | ⊥ | ⊤ |

Figure 6:

The different extensions can be obtained from a Caminada labelling of the nodes.

Our aim in this paper is to associate with a network \mathbf{P} a modal formula $\mathbf{m}(\mathbf{P})$ of a certain modal logic (which we call **LN1**) such that the following hold:

1. The modal formula $\mathbf{m}(\mathbf{P})$ contains the nodes of the network \mathbf{P} as atomic propositions.
2. The possible worlds models of the modal formula correspond in one to one fashion to the Caminada labelling of the network.

To be specific, consider a three-point linear Kripke model of the form as in Figure 5.

(i.e. if $<$ is the accessibility relation then we have $1 < 2 < 3$).

The modal logic **LN1** is such that for the Kripke model of Figure 5 above and for any modal formula $\mathbf{m}(\mathbf{P})$ for a network \mathbf{P} such that $\mathbf{m}(\mathbf{P})$ holds in the model, only three types of assignment are possible, as in Figure 6

We named the types by 0, 1, ? to correspond to the Caminada labelling.

Any model of $\mathbf{m}(\mathbf{P})$ will give truth values to the atoms in each world and the atom will acquire a type. Thus we define a Caminada function \mathbf{f} accordingly

$$\mathbf{f}(x) = \begin{cases} 1 & \text{if } x \text{ is of type 1} \\ 0 & \text{if } x \text{ is of type 0} \\ ? & \text{if } x \text{ is of type ?} \end{cases}$$

Converseley, any Caminada labelling \mathbf{f} of \mathbf{P} can be turned into an assignment to the Kripke model of Figure 6 via the same correspondence and that will be a model of $\mathbf{m}(\mathbf{P})$. Thus the logic $\mathbf{LN1}$ and the formula $\mathbf{m}(\mathbf{P})$ contains in it the information of all Caminada labelling for \mathbf{P} and therefore all types of extensions are retrievable from it by additional considerations on the models.

The modal logic formula is obtained from general methodological considerations which have nothing directly to do with argumentation. It is not something we ‘cooked up’ to correspond to the Caminada labelling. Rather, it is a result of general approach to self-referential loops in algorithmic logic, as can be seen from the next section.

2 Methodological motivation

The modal approach was first introduced by the author in 1986, in an Imperial College technical report [1] and later published in 1990 as [2].

The problem to be solved was to associate ‘logical content’ with any Prolog program π with negation as failure, especially for programs which loop.

Example 2.1 *Consider the following four programs:*

- $\pi_1:$ • a if $\neg a$
- $\pi_2:$ • a if $\neg b$
 • b if $\neg a$
- $\pi_3:$ • a if $\neg b$
 • b if $\neg a$
 • c if $\neg b \wedge \neg e$
 • d if $\neg c$
 • e if $\neg d$
- $\pi_4:$ • a if $\neg b$
 • b
 • c if $\neg a \wedge \neg b$
 • d if $\neg c$.

The problem with such programs is that they contain loops and it is not clear what they say.

The modal idea was to associate with each program π a formula $\mathbf{m}(\pi)$ of a modal logic $\mathbf{N1}$, see [2]¹ and identify $\mathbf{m}(\pi)$ via a fixed point solution of a

¹In [2] we use the modal logic $\mathbf{N1}$ of finite trees, described in the next section. The case of argumentation networks is much simpler than general logic programs and it is easier to add an axiom of linearity to $\mathbf{N1}$ to obtain $\mathbf{LN1}$ and use the linear version for our purposes. All the fixed points machinery and theorems in [2] apply to $\mathbf{LN1}$ and so we save a lot of work in this paper.

modal equation in the logic **N1**. The machinery for doing so, including effective tractable algorithms for finding $\mathbf{m}(\pi)$ is presented, motivated and discussed at length in [2]. For our purposes it is enough to explain the idea through one simple example.

We illustrate the idea for the case of the program π_1 (a if $\neg a$), see [2, Example E4, p. 197].

Let X_1 be the logical content of the program π_1 . If we understand the modality as ‘provable’ then

- q succeeds from π_1 iff

$$X_1 \vdash \Box(X_1 \rightarrow q)$$

- q fails from π_1 iff

$$X_1 \vdash \Diamond(X_1 \wedge \sim q)$$

- q loops from π_1 iff neither the above.

Therefore we need to solve something like the following modal fixed point equation for π_1

$$X \leftrightarrow \Box[a \leftrightarrow \Diamond(X \wedge \sim a)]$$

Considerations in the paper [2] lead us (for technical reasons) to the following exact equations

$$X \leftrightarrow G[\Box\perp \vee (a \leftrightarrow \Diamond(X \wedge \sim a))] \quad (*)$$

where $Gq = q \wedge \Box q$, and \Box is the irreflexive and transitive modality of the logic **N1** yet to be introduced.

It is proved in [2] that to get a solution X it is enough to substitute \top for X in the right hand side of the equation, i.e.

$$Y = \Box(\Box\perp \vee a \leftrightarrow \Diamond \sim a)$$

is a fixed point of equation (*).

The connection with argumentation comes through the following translation of an argumentation network into a Prolog program

Definition 2.2 Translation τ .

Let \mathbf{P} be a network and let $y \in \mathbf{P}$.

1. If no arrow goes into the node y then translate $\tau(y) = y$.
2. If y_1, \dots, y_n are all the nodes with arrows into y then translate

$$\tau(y) = (y \text{ if } \bigwedge_i \neg y_i).$$

Let $\pi = \tau(\mathbf{P}) = \{\tau(y) | y \in P\}$. Then π is the associated Prolog program.

It is easy to see that for the networks $\mathbf{P}_1, \dots, \mathbf{P}_4$ of Figures 1–4 the associated Prolog programs are π_1, \dots, π_4 .

Now using the machinery of [2] we obtain a modal formula $\mathbf{m}(\pi)$ for π giving ‘logical content’ to π in the logic $\mathbf{N1}$.

This same formula considered in the logic $\mathbf{LN1}$ will give logical content to \mathbf{P} .

Of course, in the next sections we will work on \mathbf{P} directly and just help ourselves to technical lemmas from [2].

3 The modal logic $\mathbf{LN1}$

This section introduces the modal logic $\mathbf{N1}$ of [2] which we use to give modal content to logic programs and consequently to argumentation networks, through its extension $\mathbf{LN1}$. The language contains atoms, the classical connectives and a modality \Box (and hence \Diamond). The models we envisage for this logic are Kripke models of the form $(S, <, a, h)$ where $(S, <, a)$ is a finite tree with root $a \in S$ and h is the assignment, giving each atomic q a subset $h(q) \subseteq S$.

The following must be satisfied.

1. If all non-endpoints are (resp. are not) in $h(q)$ then $h(q) = S$ (resp. $h(q) = \emptyset$).

Definition 3.1 ($\mathbf{N1}$ semantics) 1. A model has the form $(S, <, a, h)$, where S is a finite set, $<$ is a transitive and irreflexive relation on S , a is the actual world and the following holds:

- (a) $(S, <, a)$ is a tree with root a .
- (b) $h(q) \subseteq S$ for each atomic q . If \bar{S} is the set of endpoints of S then
 - $(S - \bar{S}) \subseteq h(q) \Rightarrow h(q) = S$.
 - $(S - \bar{S}) \cap h(q) = \emptyset \Rightarrow h(q) = \emptyset$.

2. Satisfaction is defined in the usual way

$$t \models \Box A \text{ iff } \forall s (t < s \Rightarrow s \models A)$$

3. The model satisfies A iff $a \models A$.
4. the model is an $\mathbf{LN1}$ model if the tree is a linear chain.

Definition 3.2 Axioms for $\mathbf{N1}$

1. All substitution instances of classical tautologies
2. All substitution instances of modal $\mathbf{K4}$ axioms

- $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
- $\Box A \rightarrow \Box \Box A$

- $\vdash A \Rightarrow \vdash \Box A$

3. All substitution instances of Löb's axiom

$$\Diamond A \rightarrow \Diamond(A \wedge \Box \sim A).$$

4. Axiom for atomic q :

$$4.1. q \rightarrow \Box(\sim q \rightarrow \Box q)$$

$$4.2. \Box(\Box \perp \vee q) \leftrightarrow \Box q$$

$$4.3. \Box(\Box \perp \vee \sim q) \leftrightarrow \Box \sim q.$$

5. Additional axiom of linearity for **LN1**.

- All substitution instances of

$$\Diamond A \wedge \Diamond B \rightarrow \Diamond(A \wedge B) \vee \Diamond(A \wedge \Diamond B) \vee \Diamond(B \wedge \Diamond A).$$

Theorem 3.3 1. **N1** is complete for the proposed semantics of finite trees.

2. **LN1** is complete for the proposed semantics of linear finite chains

Proof. Well known. — see [4] ■

Definition 3.4 Let \mathbf{P} be an argumentation network. For any node y in \mathbf{P} let y_1, \dots, y_n be all the nodes attacking y (i.e. with arrows leading into y). Let $\tau(y) = y \leftrightarrow \bigwedge_i \Diamond \sim y_i$. If y has no attackers let $\tau(y) = y$.

Let GA be defined as $A \wedge \Box A$.

Let $\mathbf{m}(\mathbf{P})$ be

$$G \left(\Box \perp \vee \bigwedge_{\substack{y \in \mathbf{P} \\ y \text{ has attackers } y_i}} \left(y \leftrightarrow \bigwedge_i \Diamond \sim y_i \right) \right) \wedge \bigwedge_{\substack{z \in \mathbf{P} \\ z \text{ without attackers}}} Gz$$

Let \mathbf{P} be a network. Let $y \in \mathbf{P}$ and let $y_i, i = 1, \dots, n$ be all elements attacking y (with arrows leading into y). Let us understand X as the modal 'logical content' of \mathbf{P} . Then for any $z \in P$, X proves z if z is 'on' and if z is not 'on' then $X \wedge \sim z$ is consistent. Thus for y to be 'on', we need to have

$$y \leftrightarrow \bigwedge_i \Diamond(X \wedge \sim y_i).$$

This holds for any $y \in \mathbf{P}$.

If y is not attacked by anything then we want $X \vdash y$.

Thus we expect X to satisfy more or less the equation

$$X \equiv \bigwedge_{\substack{y \in P \\ y \text{ not attacked}}} \wedge \bigwedge_{\substack{y \in \mathbf{P} \\ y \text{ attacked}}} \left(y \leftrightarrow \bigwedge_i \diamond(X \wedge \sim y_i) \right)$$

An extensive technical discussion in [2] shows that the correct equation is the following:

$$X \equiv G \left(\square \perp \vee \bigwedge_{\substack{y \in \mathbf{P} \\ y \text{ is attacked by } y_i}} \left(y \leftrightarrow \bigwedge_i \diamond(X \wedge \sim y_i) \right) \right) \wedge \bigwedge_{\substack{y \in \mathbf{P} \\ y \text{ not attacked}}} Gy$$

A sequence of Lemmas in [2] shows (see Lemma L1, page 190), that the solution to the above equation is the formula $\mathbf{m}(\mathbf{P})$ of Definition 3.4.

Lemma 3.5 *Consider a 3 point LN1 chain model of a formula $\mathbf{m}(\mathbf{P})$ of Definition 3.4. Then each atomic q has truth values of type 0 or of type 1 or of type ? of Figure 6.*

Proof. The axioms of LN1 force the following for atomic q in the chain $1 < 2 < 3$:

- $1 \models q$ and $2 \models q \Rightarrow 3 \models q$
- $1 \models \sim q$ and $2 \models \sim q \Rightarrow 3 \models \sim q$
- $1 \models q$ and $2 \models \sim q \Rightarrow 3 \models q$.

We need to prove that the option

$$1 \models q \text{ and } 2 \models \sim q \text{ and } 3 \models \sim q$$

cannot arise.

Consider a point y such as y is being attacked. (If not then Gy is in $\mathbf{m}(\mathbf{P})$ and so y is of type 1.)

We have $G(\square \perp \vee (y \leftrightarrow \bigwedge_i \diamond \sim y_i))$ holds in the model.

Hence $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ holds at points 1 and 2.

We distinguish several cases:

1. If for any j , if $y_j = \perp$ at 3 then $\diamond \sim y_j$ is true at 1 and 2 and it plays no role in the conjunction. If for all y_j , $3 \models \sim y_j$ then $y = \top$ at 1 and 2 and y is of type 1.
2. Some y_j are \top at 3. Then $\diamond \sim y_j$ is \perp at 2 and $y = \perp$ at 2.

3. We now check whether y_j is true or false at 2. If for some j , $y_j = \top$ at 2 then $\diamond \sim y_j = \perp$ at 1 and $y = \perp$ at 1 and hence $y = \perp$ everywhere, and hence y is of type 0.

If for all j , $y_j = \perp$ at 2 then $y = \top$ at 1 and we have that $y = \top$ at 1 and $y = \perp$ at 2 and we must have by the axiom 4.1 that $3 \models y$ and y is of type 3. ■

Example 3.6 We show that the axioms are needed. Consider the \mathbf{P} of Figure 7

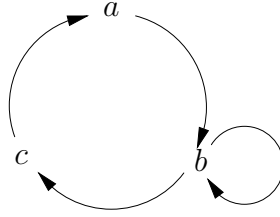


Figure 7:

We have that $\mathbf{m}(\mathbf{P})$ for this figure is:

$$G(\Box \perp \vee ((a \leftrightarrow \diamond \sim c) \wedge (b \leftrightarrow \diamond \sim a \wedge \diamond \sim b) \wedge (c \leftrightarrow \diamond \sim b)))$$

We can have the assignment of Figure 8 and $\mathbf{m}(\mathbf{P})$ still holds.

Theorem 3.7 Let \mathbf{P} be a network and let $\mathbf{m}(\mathbf{P})$ be its ‘logical content’ in the logic **LN1**.

Then there exists a one-to-one correspondence between the three chain models of $\mathbf{m}(\mathbf{P})$ and the possible Caminada labelling of \mathbf{P} . The correspondence is given by the code of Figure 6.

Proof.

Direction 1. Let h be an assignment model satisfying $\mathbf{m}(\mathbf{P})$. By Lemma 3.5 every atom node x in P gets assigned values $h(x)$ of types 0, 1, or ?.

Define a Caminada candidate function \mathbf{f} according to the types, namely $\mathbf{f}(x) = i$ iff $h(x)$ is of type i , $i \in \{0, 1, ?\}$. We now show

| | a | b | c |
|------------|---------|---------|---------|
| 3 | \perp | \top | \top |
| \uparrow | | | |
| 2 | \perp | \perp | \perp |
| \uparrow | | | |
| 1 | \top | \top | \top |

Figure 8:

that \mathbf{f} satisfies the conditions of a Caminada function of Definition 1.1.

- 1 If x is an initial point then Gx is a conjunct of $\mathbf{m}(P)$, therefore $h(x)$ is of type 1 and hence $\mathbf{f}(x) = 1$.
- 2 If y_1, \dots, y_n are all nodes with arrows to y , then we have the conjunct $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ in $\mathbf{m}(\mathbf{P})$ in the clause

$$G(\Box \perp \vee \bigwedge_y (y \leftrightarrow \bigwedge_i \diamond \sim y_i))$$

This means that at nodes 1 and 2 of the chain $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ must hold. Assume all y_i are of type 0, then $y = \top$ at 2 and 1. Hence by axiom 4.2, $y = \top$ also at node 3 and y is of type 1. This means that if all y_i are ‘out’ then y is ‘in’.

Assume one of y_i is of type 1. Then for this y_i , $\diamond \sim y_i$ is \perp at nodes 1 and 2. Hence y is false at 1 and 2 and by axiom 4.3, $y = \perp$ at node 3. Hence y is of type 0.

Now assume all of y_i are either of type 0 or of type ?, and at least one of y_i say y_1 is of type?. The y_i of type 0 have no influence on y because $\diamond \sim y_i$ is \top at nodes 1 and 2. The crucial nodes are the nodes like y_1 which are of type ?. This means that $1 \models y_1, 2 \models \sim y_1, 3 \models y_1$. Thus $\diamond \sim y_1 = \perp$ at 2 and $\diamond \sim y_1 = \top$ at 1.

This holds for any type y_i of type ?. Thus $y = \top$ at 1 and $y = \perp$ at 2. Hence by axiom 4.1, $y = \top$ at 3 and hence y is of type ?.

Direction 2. Let \mathbf{f} be a Caminada function Define a model by assigning values to the propositions according to the code of Figure 6. We claim $\mathbf{m}(\mathbf{P})$ holds in this model.

First all nodes y without arrows into them are assigned type 1 and hence Gy holds. For the other nodes we must check the formula

$$G(\Box \perp \vee \bigwedge_y (y \leftrightarrow \bigwedge_i \diamond \sim y_i))$$

y has arrows leading to it

and show it holds in the model.

We distinguish several cases. Assume all y_i are of type 0. This means $\mathbf{f}(y_i) = 0$ for all y_i . So $\diamond \sim y_i$ is \top at nodes 1 and 2. But since all $\mathbf{f}(y_i) = 0$ we get that $\mathbf{f}(y) = 1$ and hence $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ holds at 1 and 2 as required.

If one of y_i is of type 1, this means $\diamond \sim y_i$ is \perp at 1 and 2. Hence $\bigwedge_i \diamond \sim y_i$ is \perp at 1 and 2. But also since $\mathbf{f}(y_i) = 1$, we get $\mathbf{f}(y) = 0$ and so y is of type 0 and $y = \perp$ at 1 and 2.

| | Type 1 | Type 0 | Type ? | Type?* |
|---|--------|--------|--------|--------|
| 3 | ⊤ | ⊥ | ⊤ | ⊥ |
| ↑ | | | | |
| 2 | ⊤ | ⊥ | ⊥ | ⊥ |
| ↑ | | | | |
| 1 | ⊤ | ⊥ | ⊤ | ⊤ |

Figure 9:

Hence $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ holds at 1 and 2.

Now assume that all y_i are either of type 0 or type ?. This means $\mathbf{f}(y_i)$ is either 0 or ?. Assume that at least one y_i is of type ? (i.e. $\mathbf{f}(y_i) = ?$). Then $\mathbf{f}(y) = ?$ and we have $y = \top$ at 1 and $y = \perp$ at 2. Let us check whether $\bigwedge_i \diamond \sim y_i$ is \perp at and \top at 2. Since all y_i are of type ? or \perp with at least one y_i of type ?, the type ? y_i will be \top at node 2 and hence $\diamond \sim y_i = \perp$ at node 1 and hence $y = \perp$ at node 1.

On the other hand all y_i are \perp at node 3 and hence $\bigwedge_i \diamond \sim y_i$ is \top at node 2. This shows that $y \leftrightarrow \bigwedge_i \diamond \sim y_i$ is \top at nodes 1 and 2.

Thus the above argument shows that

$$G(\Box \perp \vee \bigwedge_u (y \leftrightarrow \bigwedge_i \diamond \sim y_i))$$

holds at all nodes 1, 2, and 3.

This completes the proof of the theorem. ■

Remark 3.8 (Expressive power of the modal approach) *We showed correspondence between the Caminada labelling and the 3 chain models of $\mathbf{m}(\mathbf{P})$, for the logic $\mathbf{LN1}$. Our modal logic approach is more powerful than the Caminada labelling. We can tell in more detail why a node gets value ?.*

The key is axiom 4.1. We adopted this axiom

$$q \rightarrow \Box(\sim q \rightarrow \Box q)$$

in order to get correspondence between models an Caminada labelling. We are however better off without it. If we abandon this axiom we can take any model of $\mathbf{m}(\mathbf{P})$ and using it to give a better labelling.

Consider Figure 9

We have added a new type for ‘unknown’, type (?). This new type is allowed because we abandoned axiom 4.1*

To explain, consider and compare Figure 7 and Figure 10. Figure 10 has the $\mathbf{m}(\mathbf{P})$ as

$$G(\Box \perp \vee ((a \leftrightarrow \diamond \sim c) \wedge (b \leftrightarrow \diamond \sim a) \wedge (c \leftrightarrow \diamond \sim b)))$$

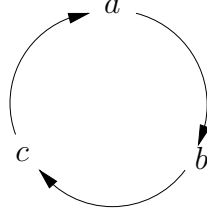


Figure 10:

This formula does not allow for any assignment of type $?^*$. While the formula for Figure 7, as given in Example 3.6, does allow type $?^*$ for b . So by the models of our $\mathbf{m}(\mathbf{P})$, we can classify better our networks, because we can distinguish between Figure 7 and 10.

Caminada labelling will give a, b, c of Figure 10 all $?$ (unknown). Same as to a, b, c of Figure 7. There is only one Caminada labelling to these figures so we cannot make distinctions between the figures using other Caminada labellings. Therefore Caminada labelling has less expressive power with regards to why nodes are unknown. We can also use chains of length $n > 3$ as models and have many more types for unknown which can be more sensitive to how the network is.

The reader might wish to know what is the intuitive meaning of a type $?^*$ label. The answer is this: If y is of type $?^*$ then y never attacks alone, i.e. $\forall x[y \text{ attacks } x \rightarrow \exists z \neq y(z \text{ attacks } x)]$.

Also if x is attacked only by type $?^*$ elements, then x is ‘on’.

The new Caminada–Gabbay rule should be (for this case) as follows:

- If y_1, \dots, y_n attack y and each y_i is either of type 0 or type $?$ or type $?^*$, and at least one of y_i is of type $?$ then y is of type $?$.

Remark 3.9 (Further advantages of the modal approach)

1. The modal approach associates a modal formula $\mathbf{m}(\mathbf{P})$ with every argumentation network \mathbf{P} . Thus we can define the concept of one network \mathbf{P}_2 being a logical extension of another network \mathbf{P}_1 . We simply say that for all x $\mathbf{m}(\mathbf{P}_1) \vdash x$ implies $\mathbf{m}(\mathbf{P}_2) \vdash x$ or equivalently $\mathbf{m}(\mathbf{P}_2) \vdash \mathbf{m}(\mathbf{P}_1)$.

There is no simple way of defining this concept directly in network terms.

2. Another option available to us is the logical characterisation of extensions. We saw that following Definition 1.1 that the different labelling options \mathbf{f} for \mathbf{P} , corresponds to all the models for $\mathbf{m}(\mathbf{P})$ and that the various extensions can be characterised by properties of \mathbf{f} and the sets $E_{\mathbf{f}}^1, E_{\mathbf{f}}^0$ and $E_{\mathbf{f}}^?$. The grounded extension for example, corresponds to minimal (among all \mathbf{f}) $E_{\mathbf{f}}^1$ and $E_{\mathbf{f}}^0$ and is unique. Therefore it can be characterised using the modal logic **LN1** as the set of all literals $z = x$ and $z = \sim x$ such that $\mathbf{m}(\mathbf{P}) \vdash \Box z$.

The other extensions such as the preferred extensions or the semi-stable extensions are not unique and need some additional language capability extending LN1 to be expressed.

References

- [1] D. M. Gabbay. Modal provability foundations for negation by failure. Imperial College Technical Report, 86/4, 1986.
- [2] D. M. Gabbay. Modal provability foundations for negation by failure, in *Extensions of Logic Programming*, P. Schroeder-Heister, editor, pp. 179–222, Vol 475 of LNCS, Springer-verlag, 1990.
- [3] T. Przymusiński. Well founded semantics coincides with three valued stable semantics. Special issue of *Fundamenta Informatica* on Non-monotonic reasoning. 13(4), W. Marek, ed. 445–464, 1990.
- [4] D. M. Gabbay. *Semantical Investigations in Modal and Tense Logics*. Synthese, Volume 92, D. Reidel, 1976.
- [5] M. Caminada. A gentle introduction to argumentation semantics. Lecture material, Summer 2008.
- [6] P. M. Dung. On acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77, 321–357, 1995.
- [7] P. M. Dung. An argumentation theoretic foundation for logic programming. *Journal of Logic Programming*, 22(2), 151–171, 1995.
- [8] H. Barringer, D. Gabbay and J. Woods. Temporal dynamics of argumentation networks. In *Mechanising Mathematical Reasoning*, D. Hutter and W. Stephan, eds., pp. 59–98. LNCS 2605, Springer, 2005.