

## Modal and temporal argumentation networks<sup>†</sup>

H. Barringer<sup>a</sup>, D.M. Gabbay<sup>b,c,d,\*</sup> and J. Woods<sup>e</sup>

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

(Received 21 October 2011; final version received 15 June 2012)

The traditional Dung networks depict arguments as atomic and study the relationships of attack between them. This can be generalised in two ways. One is to consider various forms of attack, support, feedback, etc. Another is to add content to nodes and put there not just atomic arguments but more structure, e.g. proofs in some logic or simply just formulas from a richer language. This paper offers to use temporal and modal language formulas to represent arguments in the nodes of a network. The suitable semantics for such networks is Kripke semantics. We also introduce a new key concept of usability of an argument. This is the beginning of a continuing research for adding contents to the nodes of an argumentation network. This research will allow us to address notions like 'what does it exactly mean for a node to attack another' or 'what does it mean for a network to be consistent' or 'can we give proper proof rules to manipulate networks', and more.

**Keywords:** modal logic; temporal logic; argumentation networks; Kripke models

### 1. Introduction and orientation

This section will look at modal and temporal logics in a way which is compatible with argumentation networks. This will allow us to understand our options in introducing modal and temporal argumentation networks.

We adopt a context view of modal logic. Assume we are talking about various contexts, which we denote by  $w_1, w_2, \dots$ , and we discuss a finite number of atomic facts,  $Q = \{q_1, \dots, q_n\}$ . We write  $w \models q$  to mean that  $q$  holds in the context  $w$ . Modal logic allows us to talk from within a context  $w$  about other-related contexts. Let  $R$  be an accessibility relation on the set  $W$  of contexts, so we write  $wRw'$  to mean  $w'$  is accessible to  $w$ .

If we add the modal connective  $\diamond$  to our language, we can write

- $w \models \diamond q$  iff for some  $w'$ , such that  $wRw'$ , we have  $w' \models q$ .

This gives us basic modal logic.

The first question we ask is where does the relation  $w \models q$  come from? In traditional modal logic, this is arbitrary.

A traditional modal model (for the logic **K**) comes as a triple  $(W, R, \models)$ , where  $W$  is the set of worlds (contexts),  $R \subseteq W \times W$  is the accessibility relation and  $\models$  is the relation which tells us which atoms  $q \in Q$  hold at which world (i.e. when  $w \models q$  holds).  $\models$  is arbitrarily given. One can

---

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

<sup>†</sup>This paper is an expanded version of Barringer and Gabbay (2010). Research done under ISF project: Integrating Logic and Network Reasoning.

modify abstract modal logic and add more details about the origins of  $\models$ . For example, we can imagine that each context  $w$  has a database  $\Delta_w$  associated with it and we write

$$w \models q \text{ iff } \Delta_w \vdash q$$

If we do this, we no longer have the semantics of arbitrary modal logic  $\mathbf{K}$  and we may get some special versions/extensions of  $\mathbf{K}$ .

In the case of argumentation, we may associate with each context  $w$  an argumentation network  $N_w = (Q, \rho_w)$ , where  $Q$  is the set of atoms and  $\rho_2 \subseteq Q \times Q$  is the attack relation. We need to choose an extension  $E_w \subseteq Q$  calculated properly using traditional Dung rules from the attack relation  $\rho_w$ . We thus have

$$w \vdash q \text{ iff } q \in E_w.$$

Thus in this approach, the argumentation networks are vehicles for defining  $\models$ . This has intuitive sense. In the world  $w$ , there is a local perception of attack  $\rho_w$  about the facts of  $w$  and a debate resulting in an extension  $E_w$ , and this extension tells us what facts hold in the context  $w$ . So, really we should write

$$N_w = (Q, \rho_w, E_w).$$

If we adopt this approach, we will have a problem. Modal logic allows us to use expressions like  $\diamond q$  at any world  $w$ . So, we need to allow  $\diamond q$  to enter the argumentation considerations taking place at world  $w$ .

Let  $Q_\diamond = Q \cup \{\diamond q \mid q \in Q\}$ . We have

$$N_w = (Q_\diamond, \rho_w, E_w)$$

with

$$\rho_w \subseteq Q_\diamond \times Q_\diamond$$

and  $E_w$  an extension,

$$E_w \subseteq Q_\diamond.$$

We ask how is this extension calculated?

Suppose we have in the world  $w$ , the case that  $\diamond q$  attacks  $p$  (i.e.  $\diamond q \rho_w p$ ) and  $\diamond q$  is not attacked by any other argument. We would, therefore, expect that  $\diamond q \in E_w$ , i.e.  $w \models \diamond q$  holds.

On the other hand, we also expect that for some  $w'$  such that  $wRw'$  we have  $w' \models q$ , i.e. that  $q \in E_{w'}$ . However, such a  $w'$  may not exist!

We get an internal incompatibility in the system. Obviously we need an internal device to deal with this either an additional compatibility postulate or by something else. We solve this problem by introducing the concept of usability. We have a usability function  $h_w$  and we have in this case  $h_w(\diamond q) = 0$ , i.e. we cannot use it in the considerations of finding an extension  $E_w$ .

Of course, there is independent motivation to the concept of usability. Its introduction is not just technical.

In the above considerations, the dominant view was that of modal logic, and the argumentation network view was auxiliary; it provided a means of defining  $\models$ . Can we look at the entire possible world system as one big argumentation network? Put differently, can we view  $(W, R, N_w)$ ,  $w \in W$ , as one big argumentation network?

This is possible to do using auxiliary arguments which eliminate  $R$  and  $W$ . We first explain this by example. Consider Figure 1. There are three worlds  $w, w_1, w_2$ . We have  $wRw_1$  and  $wRw_2$ . We assume the grounded extension in each world.  $\diamond q$  is not attacked in  $w_1$ . It says that it is possible to have an accessible world in which  $q$  holds. So, we must have either  $q$  in the grounded extension of world  $w_1$  or  $q$  in the grounded extension of world  $w_2$ .

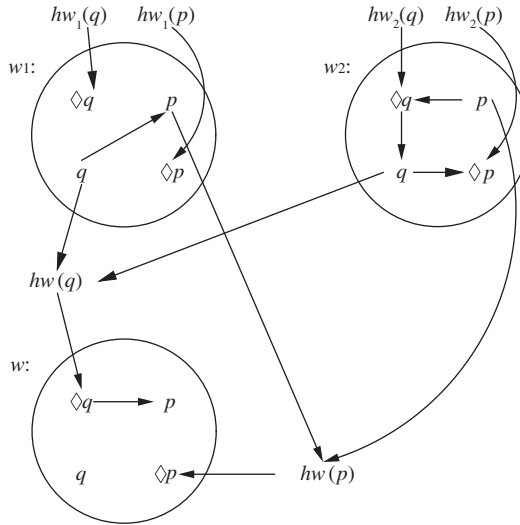


Figure 1. Example for eliminating R and W.

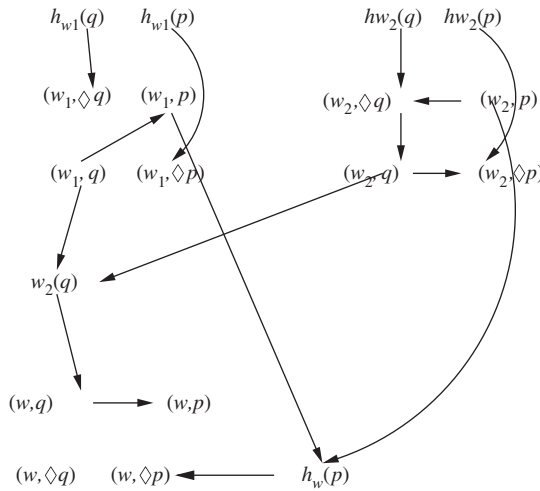


Figure 2. Result after R and W eliminated from Figure 1.

If this holds, we say  $\diamond q$  is usable in  $w$ , otherwise not. This usability is implemented in the object level by adding the point  $h_w(q)$  as a meta-point outside the possible worlds, which point attacks  $\diamond q$  of the world  $w$ , and letting all the  $qs$  of the accessible worlds attack it. By the traditional rules of Dung argumentation, if at least one of the  $qs$  in  $w_1$  or in  $w_2$  is in, then  $h_w(q)$  is out and so  $\diamond q$  is in. Otherwise,  $\diamond q$  is out (not usable) in  $w$ . We do the same trick for  $\diamond p$ . If we want to eliminate the big circles indicating the worlds, we have to annotate the atoms by the world name. If we do that we get Figure 2, which is one big uniform argumentation network where we seek the grounded extension.

The above discussion shows what options we have in principle in formulating modal and temporal argumentation networks.

- (1) Let modal logic dominate by keeping the possible worlds separate and allow for an argumentation network to say what holds in each world.
- (2) Incorporate the modal aspects inside one big argumentation network by using auxiliary meta-arguments. This way we are compiling modal logic into argumentation. This is technically possible to do, even possible to do nicely, in view of the results in Gabbay’s (2011) paper, showing that argumentation is equivalent to classical logic in a nice way.
- (3) In the temporal logic case, there is a simpler and more immediate way of defining a temporal logic network. We simply time stamp each argument with a moment of time or an interval of time, these being the times when this argument is considered usable. Thus, a network has the form  $(T, \rho, \tau)$ , where  $T$  is the set of arguments,  $\rho$  is the attack relation, and  $\tau$  is a function,

$$\tau : T \cup \rho \Rightarrow \mathbb{R},$$

where  $\tau(x) =$  moments of time, where  $x$  is usable.

There is a very simple model, used in 2005 in Barringer, Gabbay, and Woods (2005), in connection with temporally varying numerical strength of arguments.

In itself, the model is too simple and does not offer much, but one can easily add natural structure to it to indicate evolution over time, as done in Barringer et al. (2005) and Abraham, Gabbay, and Schild ((2011a, b). In Barringer et al. (2005), the change in time was of strength of argument and this can influence the argument’s attack capabilities and (Abraham et al. 2011a) the time stamping was used to resolve loops.

**2. Introducing the global meta-level approach to temporal networks: concept of usability**

There are several good reasons why we should consider modal and temporal argumentation networks

- (1) *Temporal facts as arguments.* Past facts or future scheduled events can also be used as arguments for the present. An argument against the trustworthiness of a person may be the facts of past betrayals. An argument in favour of a higher mortgage loan may be a scheduled increment in salary next year. Unfortunately, this does not work with UK banks. An argument against a higher mortgage loan may be the possibility of redundancy in the future. Figure 3 is an example of how a scheduled redundancy exercise in the near future can be used as an argument against a high mortgage loan now, where  $d$  means the event of redundancy and  $m$  the general argument in favour of a mortgage. We shall discuss later why it is not reasonable to encapsulate ‘Future  $d$ ’ as a single argument  $c$  attacking  $m$  now. We lose structure this way (compare with how we lose structure in

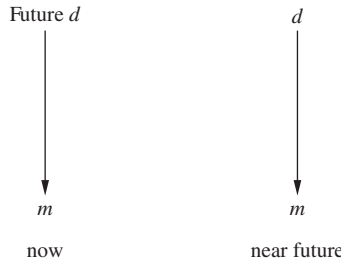


Figure 3. Future used as argument against present.

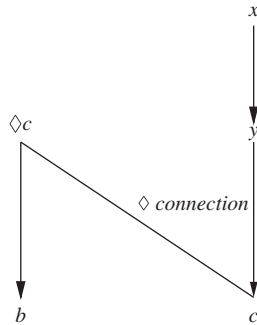


Figure 4. Fibring arguments.

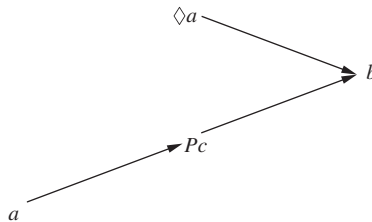


Figure 5. Using a past argument to attack a present argument.

propositional logic in case  $\forall xA(x) \vdash A(\alpha)$ . We need predicate logic to go into the structure of the sentences.

- (2) *Fibring arguments.* Arguments from one domain may be brought into another domain. For example, expert medical arguments may be brought in as a package into a legal argument. This may be best treated in the context of modal logic (bringing information from the medical world into the legal world), where  $\diamond x$  means bring in information  $x$  from another world, i.e. domain. See Figure 4. Let  $b$  be the legal argument to commit the accused to a 1-year prison sentence for tax evasion. Let  $c$  be the medical argument that the accused has cancer. This medical argument attacks  $b$  in the legal world. A hefty fine is more appropriate.  $c$  is part of a medical network of arguments and  $c$  emerges among the winning arguments of that network. Figure 4 illustrates the situation. Of course, in the legal world  $\diamond c$  might be attacked as unacceptable evidence on the basis of some procedural errors in putting in forward (not shown in diagram).
- (3) *Future arguments.* The possibility that an argument  $a$  may be able to defeat another argument  $b$ . We denote this by  $\diamond a$ . Such possibilities are central to threats and negotiations' arguments where various future scenarios are discussed. For example, do not ask for more than a 10% salary settlement as it will never be approved by the executives – there may be strong fairness arguments for claiming 10% but pragmatically it will not be affordable and thus will not get approved.
- (4) *Past arguments.* We can use the fact that an argument  $c$  was potent in the past (denoted by  $Pc$ ) to attack another current argument. Figure 5 is an example of such a configuration, where  $\diamond a$  indicates that argument  $a$  is possible and  $Pc$  indicates that argument  $c$  was considered in the past, but maybe is no longer taken seriously now, yet the fact that it was a serious argument in the past is sufficient to defeat  $b$ . For example, a mother might say 'I have always cared for you, you cannot abandon me now'.

For example, a female employee may threaten with a possible argument claiming harassment. It may be that one cannot argue harassment now but it is not clear what the circumstances would look like when reviewed in the future. So,  $\diamond$ harassment may have some force. We have had many such arguments when UK law was expected to be overruled by EU law. Many  $\diamond$ (EUlaw) arguments were already defeating local UK arguments even before the EU law came into force in the UK.

Negotiations always involve evaluation of future scenarios of possible arguments and counter arguments and the possibilities of certain scenarios may be a strong argument at present.

Arguments from the point of view of tradition have always been successful in the past, e.g. we have always accepted the A'level standard as an appropriate university entrance qualification, so we continue to do so, even though many will argue that the level has dropped. We can have our doubts about the value of tradition now but yet an argument of the form 'but it has always been the case that x' may still win out.

Any use of precedent is also akin to this form.

The basic notion is that of admissible extension. This is a subset  $E \subseteq S$  such that

- (1)  $E$  is conflict free, i.e. for no  $x, y \in E$  can we have  $(x, y) \in R$ .
- (2)  $E$  is self-defending, i.e. for all  $x, y \in S$ , if  $x \in E$  and  $(y, x) \in R$ , then there exists a  $z \in S$  such that  $(z, y) \in R$ .

The above discussion suggests that we introduce the concept of usability of arguments. We may have at a certain time or at a certain context some arguments that are talked about and are available in some real sense, but these arguments cannot be used for a variety of reasons. The formal presentation of such arguments can be to introduce them into the network but label them as unusable through a usability function  $h$ . If  $x$  is an argument, then  $h(x) = 1$  means it is usable and  $h(x) = 0$  means it is not. The reader may ask why do we want to introduce them at all if they are not usable? Well, in the context of modal and temporal logics, it makes sense to talk about them. Maybe they were usable, maybe they will be usable or are possibly but not necessarily usable, or should have been usable, etc. We give several examples.

*Example 2.1* (The catholic super administrator) A UK university, operating an equal opportunities policy, advertises for a faculty administrator. There is a shortlist of three candidates and, because of a special request from one candidate, the interview date is moved.<sup>1</sup>

The top two candidates are: a woman aged 42, who knows 15 languages and 10 computer languages and has a PhD in economics and business administration from Harvard. She has lots of experience working for government administration; the other candidate is a man of similar age, but not with as strong a background as the lady.

There is an argument for hiring the lady candidate: she is the best!

There is an argument against hiring the lady candidate: she is Catholic, aged 42, recently married, and will probably waste no time in starting a family.

This latter argument is a strong subjective argument, which, following the proper procedures, cannot be used. Indeed, one cannot mention it, let alone even think it!  $h$  (this argument) = 0.<sup>2</sup>

*Example 2.2* (The rape) A girl complained she was raped by a man late at night in the street. The man claimed that she gave him reason to take the view that she was willing and available. The entire incident was filmed, video and audio, by a CCTV camera.

However, this camera was installed without a licence and hence, because of a legal technicality, any evidence from the CCTV is not admissible. The evidence from the CCTV clearly and

unambiguously defeated the claim of the man, but because of its inadmissibility, the jury was instructed to accept the man's claim.

In both cases, we present the network in the traditional form  $(S, R)$ , where  $S$  is the set of arguments and  $R$  is the attack relation, but with both usable and unusable arguments included, where those that are usable are marked via a function  $h$ .  $h(a) = 0$  means we cannot use argument  $a$ .  $h(a) = 1$  means we can use the argument.

It is important to note that unusability is temporary and can change. Circumstances can change, the law can change, new arguments can be brought forward and what was unusable can become usable.

- (1) *Unusability due to defeat.* Figure 4 can be an example of unusable argument. The notation  $\diamond c$ , wants to bring the cancer argument from another network, the medical network into the present network, the legal one. In the figure,  $c$  is a winning argument in the medical network and it is attacked by  $y$  in the figure but is defended by  $x$ . However, it is quite possible in a complex cross-network situation, that we have a  $\diamond z$  such that in the appropriate network for  $z$ ,  $z$  is defeated. In this case, we can view  $\diamond z$  as unusable. Again, this is not permanent and may change.
- (2) *Unusability due to secrecy.* It is quite possible that an argument  $a$  is defeated by an argument  $a^*$  which cannot be recorded explicitly in the system. In this case, it may be convenient not to mention  $a^*$  and to simply mark  $a$  as unusable.

### 3. Temporal networks: formal considerations

We need to define the formal machinery and distinctions allowing us to put in context our approach to modal and temporal argumentation networks. So, we define some basic notions in this section and move on to the Kripke models in the next section.

DEFINITION 3.1 (General labelled networks)

- (1) A general labelled network has the form

$$\mathbb{N} = (T, \rho, \mathbf{l}, \mathbf{f}),$$

where  $T$  is a set of nodes and  $\rho \subseteq T \times T$  is a binary relation on  $T$ .  $\mathbf{l}$  is a labelling function on  $T \cup \rho$  giving labels from a set of labels  $\mathbb{L}$  (usually  $\mathbb{L} = \{0, 1\}$  or  $[0, 1]$ ). The label  $\mathbf{l}(t)$  for  $t \in T$  can be thought of as the strength of the node. The label  $\mathbf{l}(t, s)$  for  $(t, s) \in \rho$  can be thought of as the transmission label from  $t$  to  $s$ .

The functional  $\mathbf{f}$  is an update functional, it updates the labelling function  $\mathbf{l}$  to a new one  $\mathbf{f}(\mathbf{l})$ .  $\mathbf{f}$  is a pair of functions,  $\mathbf{f}_1, \mathbf{f}_2$ , which operate on multisets of elements to give a new element. For example, the function 'maximum' or the function 'take the sum of' is such a function. We write the value  $\mathbf{f}_i(x, y, z, \dots)$ , where  $(x, y, z, \dots)$  is a sequence or a multiset. Given a node  $t$ , let  $\rho(t)$  be  $\{s \mid s\rho t \text{ holds}\}$ . Then for any  $t$ , let

$$\mathbf{f}(\mathbf{l})(t) = \mathbf{f}_1(\mathbf{l}(t), \mathbf{l}(s), \mathbf{l}(s, t), s \in \rho(t))$$

$$\mathbf{f}(\mathbf{l})(s, t) = \mathbf{f}_2(\mathbf{l}(t), \mathbf{l}(s), \mathbf{l}(s, t))$$

be new labels at  $t$ , and  $\text{at}(s, t)$  given by the functional  $\mathbf{f}$ .  $\mathbf{f}$  depends on  $\rho$  and  $\mathbf{l}$ , and on the labels and transmission labels, as depicted in Figure 6.

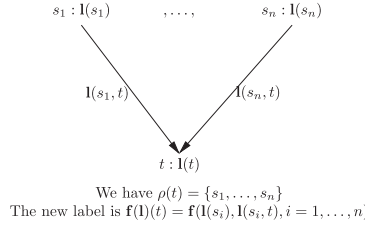


Figure 6. A general labelled network.

The way  $\mathbf{f}$  is calculated is not described here. The reader can compare later in the section below, where we give some examples of algorithms for  $\mathbf{f}$  in terms of  $\rho$  and  $\mathbf{l}$ .

- (2)  $\mathbf{f}$  can be used for successive updating of the labelling of our network.

We define  $\mathbf{l}_m$  for  $m \geq 0$  by induction on  $m$ .

*Step 0* :  $\mathbf{l}_0 = \mathbf{l}$ .

*Step  $m + 1$*  :  $\mathbf{l}_{m+1} = \mathbf{f}(\mathbf{l}_m)$

Let  $?$  be a fixed label in  $\mathbb{L}$ . We can define  $\mathbf{l}_\infty$  using  $?$  by letting  $\mathbf{l}_\infty(x) = y$ , if for some  $k$  we have  $\mathbf{l}_m(x) = y$  for all  $m \geq k$ , and otherwise let  $\mathbf{l}_\infty(x) = ?$  if  $x$  is either a node  $t \in T$  or a connection  $(s, t) \in \rho$ .

We now have the machinery to look at argumentation networks and we use the Caminada labelling for them (Caminada and Gabbay 2009; Caminada 2011).

#### DEFINITION 3.2 (Argumentation model)

- (1) Let  $\mathcal{C}$  be the language of the classical propositional calculus with atoms  $\mathcal{Q}$  and connectives  $\neg, \wedge, \vee, \rightarrow, \top, \perp$ . Where  $\mathcal{Q}$  is the set of atomic arguments.
- (2) An argumentation model has the form  $\mathbb{N} = (\mathbb{F}, \rho, h)$  where  $\mathbb{F}$  is a set of formulas,  $\rho$  is a binary relation on  $\mathbb{F}$  and  $h$  an assignment of truth values to the atoms  $\mathcal{Q}$ . We can view  $h$  as a subset  $h \subseteq \mathcal{Q}$ , and think of it as the set of usable arguments.
- (3) Given  $h$ , we can assign usability values to the formulas of  $\mathbb{F}$  using the traditional truth table. We write  $h(A)$  as the value of a wff  $A$  under  $h$ .  $h$  can now be regarded as a subset of  $\mathbb{F}$ .
- (4) A network is atomic iff  $\mathbb{F} \subseteq \mathcal{Q}$ .
- (5) Note that  $h$  gives initial usability values which are not necessarily permanent and may change in the course of the recursive evaluation, see Definition 3.3.

DEFINITION 3.3 Let  $\mathbb{N} = (\mathbb{F}, \rho, h)$  be an argumentation model. We define an algorithm for extracting winning arguments out of  $\mathbb{N}$  as follows. The definition is by levels. We define  $h_m(A)$  for  $A \in \mathbb{F}$  by induction on  $m$  (compare with Definition 3.1).

*Level 0*

$$h_0(A) = h(A).$$

*Level  $m + 1$*

Let  $A \in \mathbb{F}$  and let  $\rho(A) = \{B_1, \dots, B_k\}$  be all formulas  $B$  of  $\mathbb{F}$  such that  $B\rho A$  holds. These are the formulas which attack  $A$  according to  $\rho$ . There are several possibilities

- (1)  $h_m(B) = 0$  for all  $B \in \rho(A)$ . In this case, let  $h_{m+1}(A) = 1$ .
- (2) For some  $B \in \rho(A)$ ,  $h_m(B) = 1$ . In this case, let  $h_{m+1}(A) = 0$ .
- (3)  $\rho(A) = \emptyset$ , in which case, let  $h_{m+1}(A) = 1$



Let  $\pi$  be the operation which defines  $h_{m+1}$  out of  $h_m$ , i.e.  $h_{m+1} = \pi h_m$ . Of course,  $\pi$  depends on  $\rho$ . To be more explicit about the role of  $\pi$ , assume  $H$  is a function giving  $\{0, 1\}$  values to all elements of  $\mathbb{F}$ . Using  $\rho$  and rules (1)–(3) above we can transform  $H$  into  $H'$ . We write  $H' = \pi H$ .<sup>3</sup>

*Level  $\infty$*

Let  $h_\infty(A) = y \in \{0, 1\}$  iff for some  $k$ ,  $h_m(A) = y$  for all  $m \geq k$ .

Let  $h_\infty(A) = ?$  (undefined) otherwise.

$h_\infty$  is called the *BG* labelling of  $\mathbb{F}$ .

**DEFINITION 3.4** (Caminada labelling) Let  $(\mathbb{F}, \rho)$  be an atomic network. A Caminada labelling on  $\mathbb{F}$  is a function  $\lambda$  giving values in  $\{0, 1, ?\}$  satisfying the following:

- (1) if  $\rho(x) = \emptyset$ , then  $\lambda(x) = 1$
- (2) If for some  $y \in \rho(x)$ ,  $\lambda(y) = 1$ , then  $\lambda(x) = 0$ .
- (3) If for all  $y \in \rho(x)$ ,  $\lambda(y) = 0$ , then  $\lambda(x) = 1$ .
- (4) If for some  $y \in \rho(x)$ ,  $\lambda(y) = ?$  and for no  $y \in \rho(x)$  do we have  $\lambda(y) = 1$ , then  $\lambda(x) = ?$

**LEMMA 3.5** Let  $\mathbb{N} = (\mathbb{F}, \rho, \lambda)$  be an atomic network with the Caminada labelling  $\lambda$ . Then, there exists an assignment  $h_0$  such that  $h_\infty = \lambda$ .

*Proof* Let  $h^+(x) = 1$  if  $\lambda(x) = 1$  or  $\lambda(x) = ?$

Let  $h^+(x) = 0$  if  $\lambda(x) = 0$ .

Let  $h^-(x) = 1$  if  $\lambda(x) = 1$ .

Let  $h^-(x) = 0$  if  $\lambda(x) = 0$  or  $\lambda(x) = ?$

Let  $h_0 = h^+$ .

We now prove

- (1) If  $h_m = h^+$ , then  $h_{m+1} = h^-$ .
- (2) If  $h_m = h^-$ , then  $h_{m+1} = h^+$ .

Assume  $h_m = h^\pm$ . We calculate  $h_{m+1}(x)$ ,  $x \in \mathbb{F}$ , and show  $h_{m+1} = h^\mp$ .

If  $\lambda(x) = 1$ , then either  $\rho(x) = \emptyset$  or for all  $y \in \rho(x)$ ,  $\lambda(y) = 0$ . In this case,  $h_m(y) = 0$  and so  $h_{m+1}(x) = 1$ .

If  $\lambda(x) = 0$ , then for some  $y \in \rho(x)$ ,  $\lambda(y) = 1$ . In this case,  $h_m(y) = 1$  and so  $h_{m+1}(x) = 0$ .

If  $\lambda(x) = ?$ , then  $\rho(x) \neq \emptyset$  and for some  $y \in \rho(x)$ ,  $\lambda(y) = ?$  and for none of the other  $y \in \rho(x)$  do we have  $\lambda(y) = 1$ . So, let  $\{y_1, \dots, y_k, y_{k+1}, \dots, y_r\} = \rho(x)$ , with  $k \geq 1$ ,  $\lambda(y_j) = ?$ ,  $1 \leq j \leq k$  and  $\lambda(y_{k+1}) = \dots = \lambda(y_r) = 0$ .

Clearly if  $h_m = h^\pm$ , then  $h_m(y_{k+1}) = \dots = h_m(y_r) = 0$ .

If  $h_m = h^+$  then  $h_m(x) = 1$  and  $h_m(y_1) = \dots = h_m(y_k) = 1$  and hence  $h_{m+1}(x) = 0$ .

This shows that  $h_{m+1} = h^-$ , since  $x$  was arbitrary.

If  $h_m = h^-$  then  $h_m(x) = 0$  and  $h_m(y_1) = \dots = h_m(y_k) = 0$  and so for all  $y \in \rho(x)$ ,  $h_m(y) = 0$  hence  $h_{m+1}(x) = 1$ .

Again since  $x$  was arbitrary, we get  $h_{m+1} = h^+$ .

So, if we start with  $h_0 = h^+$ , we get  $h_{2m} = h^+$ ,  $h_{2m+1} = h^-$  and so  $h_\infty = \lambda$ . ■

**LEMMA 3.6** The converse of the previous lemma does not hold. Not every  $h_\infty$  is a Caminada labelling.

*Proof* Consider the network in Figure 7

Start with  $h_0(a) = h_0(b) = 1$ ,  $h_0(c) = h_0(d) = 0$ ,  $h_0(e) = 0$ . We have  $h_1(a) = h_1(b) = 0$ ,  $h_1(c) = h_1(d) = 1$ ,  $h_1(e) = 0$ .

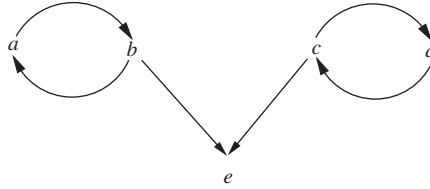


Figure 7. Oscillation and attack.

We also have

$$h_{2m} = h_0, h_{2m+1} = h_1.$$

Thus,  $h_\infty(a) = h_\infty(b) = h_\infty(c) = h_\infty(d) = ?$  and  $h_\infty(e) = 0$ .

The Caminada labelling rules do not allow for  $\lambda = h_\infty$ . ■

*Remark 3.7*

- (1) The reason we could provide the example in Figure 7 is that  $h$  gave value 1 to the loop  $(a, b)$  and value 0 to the loop  $(c, d)$ . So, as the values in the loop oscillated, there was always one loop which attacked  $e$ . If all loops were to oscillate synchronously,  $h(e)$  would have oscillated as well.

How can we overcome this? We can use ultrafilters to get an exact value out of the oscillation. We need some concepts

Let  $\mathbb{N}$  be the set of natural numbers. A family of subsets  $\mathbb{U}$  of numbers is called an ultrafilter if the following holds

- (a)  $\mathbb{N} \in \mathbb{U}, \emptyset \notin \mathbb{U}$
- (b) If  $X, Y \in \mathbb{U}$ , then  $X \cap Y \in \mathbb{U}$
- (c) either  $X$  or  $\mathbb{N} - X$  is in  $\mathbb{U}$ .

$\mathbb{U}$  says which sets are ‘big’.

We also note that there exists an ultrafilter  $\mathbb{U}$  such that all co-finite sets are in  $\mathbb{U}$ .

We now give an alternative definition of  $h_\infty$ . Call it  $h_\omega$ .

$$h_\omega(x) = 1 \text{ iff } U_x = \{m | h_m(x) = 1\} \in \mathbb{U}.$$

Let us see what happens with the example of Figure 7 if we use  $h_\omega$  instead of  $h_\infty$ . We have  $U_a = U_b = \text{all even numbers}$ .

$U_c = U_d = \text{all odd numbers}$ .

One of two sets {odd numbers, even numbers} is in  $\mathbb{U}$ . From symmetry, we can assume without loss of generality that it is the even numbers. We get

$$h_\omega(a) = h_\omega(b) = 1, h_\omega(c) = h_\omega(d) = 0.$$

So  $h_\omega$  is the same as  $h_0$  and we have nothing.

Let us try another angle.

- (2) The discrepancy with the Caminada labelling and hence with the Dung network rules seem to arise in the case where a winning argument  $x$  according to Dung gets a value  $h(x) = 0$ . Figure 8 gives two typical examples.

According to the Dung rules  $a, d, c$  are winning arguments. If  $h$  is such  $h(a) = 0$ , then  $b$  and  $e$  will be the winning arguments.

The question we ask is can we use a device which makes the two approaches compatible?

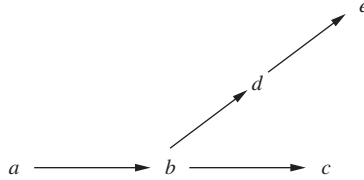


Figure 8. Example showing the discrepancy with the Caminada labelling.

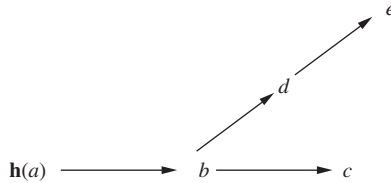


Figure 9. The real meaning of Figure 8.

Suppose we say  $a$  is not usable (i.e.  $h(a) = 0$ ) because there is an attack on  $a$ . Say  $\mathbf{h}(a)$  is the argument which attacks  $a$ .  $\mathbf{h}(a)$  is not in the network but the fact that  $a$  is attacked is recorded by  $h(a) = 0$ . There may be good reason why we do not want  $\mathbf{h}(a)$  to be explicit in the network. Maybe  $\mathbf{h}(a)$  is a different type of argument. Maybe it is a secret argument. Whatever the reason is, the real network is Figure 9.

The above trick works for this network. Does it work in general?

Given an atomic network  $(\mathbb{F}, \rho, h)$ , can we get the correct result by adding nodes  $\mathbb{F}_1 = \{\mathbf{h}(x) | x \text{ such that } h(x) = 0\}$  and letting  $\mathbb{F}' = \mathbb{F} \cup \mathbb{F}_1$  and  $\rho' = \rho \cup \{(\mathbf{h}(x), x) | \mathbf{h}(x) \in \mathbb{F}_1\}$ ? Do we have a general theorem that we can pair the winning subsets? I.e. can we have:

- (a) For any winning set  $T' \subseteq \mathbb{F}'$ ,  $T' \cap \mathbb{F}$  is a winning set of  $(\mathbb{F}, \rho, h)$ .
- (b) For any winning set  $T \subseteq \mathbb{F}$ , there exists a winning set  $T' \subseteq \mathbb{F}'$  such that  $T = T' \cap \mathbb{F}$ .

We can hope for such results only if whenever  $h$  says an argument  $x$  is out then it is out permanently, because when we insert  $\mathbf{h}(x)$  to attack  $x$  and force it out, it is out permanently. Our algorithm in Definition 3.3 and later on in the section dealing with modal and temporal logics, does not keep unusable arguments out permanently, it does bring them in depending on the attack cycles.

*Remark 3.8* (Discussion of the Dung network rules)

- (1) The discrepancy with the Caminada labelling is a serious one. The Caminada labelling is faithful to the Dung argumentation network rules, namely (Definition 3.3)
  - (a) If all attacks on a node  $x$  are defeated (are *out*), then  $x$  is *in*.
  - (b) If some attacks on a node  $x$  are *in*, then  $x$  is *out*.
  - (c) If there are no nodes attacking  $x$ , then  $x$  is *in*.

In the Dung framework, these rules are *not defeasible rules*, they are *absolute*.

So consider, for example, an argumentation network with one node and one argument  $x$ . Since nothing attacks  $x$ ,  $x$  is a winning argument. If we look at a classical model for  $x$  with  $x = 0$ , namely  $x$  is unusable for whatever reason, then the Dung rule overrides the unusability of  $x$  and  $x$  is still a winning argument.

Compare this with default logic. The default rule  $x/x$  says that if  $x$  is consistent to add then add it by default. This will not override any given data about  $x$ .

So, rules (a)–(c) are too strong when we give a model interpretation to the arguments. An argument can win even though it is unusable, simply because it is not attacked.

- (2) We call for a new critical evaluation of rules (a)–(c). We would abandon rule (c) and modify rule (a).

The proposed BG rules for a Dung network relative to an assignment or other evidence are (a\*)–(c\*) below. We call the associated update functional  $\pi^*$  (compare with  $\pi$  of Definition 3.3).

- (a\*) If all attacks on  $x$  are defeated and there is no evidence that  $x$  is unusable, then  $x$  is *in*
- (b\*) If some attacks on  $x$  are in, then  $x$  is *out*.
- (c\*) If there are no nodes attacking  $x$ , then  $x$  is *in* only if there is no evidence that it is unusable.

Jakobovits and Vermeir (1999) have already proposed that an argument which has all of its defeaters out need not necessarily be in. This view is criticised in Caminada (2006). Our (a\*) and (c\*) are in line with Jakobovits and Vermeir (1999).

- (3) Let us call an assignment  $h$  a Caminada assignment to the atoms of  $Q$  if for some Caminada labelling  $\lambda$  we have

- $h(x) = 1$  if  $\lambda(x) = 1$ .
- $h(x) = 0$  if  $\lambda(x) = 0$ .
- $\forall x(\lambda(x) = ? \text{ implies } h(x) = 1)$ .

From Lemma 3.5, we know that  $h_\infty = \lambda$  and hence Caminada assignments are compatible with Dung networks. If we restrict our Kripke model to Caminada assignments, then maybe we will have no technical discrepancies.

*Remark 3.9* The difference between BG and Caminada labelling can be appreciated by looking at the logic programming translation of a Dung network. Consider the network  $(\{a, b\}, \{(ab)\})$  (that is a network with two arguments  $a$  and  $b$  with  $a$  attacking  $b$ ). Its translation is ( $\neg$  is negation as failure)

- (1)  $b$  if  $\neg a$ .
- (2)  $a$ .

In our model, we also give usability assignments  $h(x)$  to  $x$ . So, we translate as follows into a logic programming program (we regard  $\mathbf{h}(x)$  as a new literal dependent on  $x$  attacking  $x$  by virtue of an argument showing that  $x$  is unusable) :

- (1\*)  $b$  if  $\neg a \wedge \neg \mathbf{h}(b)$ .
- (2\*)  $a$  if  $\neg \mathbf{h}(a)$ .
- (3\*)  $\mathbf{h}(x)$  if  $\neg \mathbf{zero}(x)$ , for all nodes  $x$ .
- (4\*)  $\mathbf{zero}(x)$ , for all  $x$  such that  $x = \text{usable}$ , under the assignment  $h$ .

$\mathbf{h}(x)$  and  $\mathbf{zero}(x)$  are new literals, for each node  $x$ .

Note that we use  $\neg \mathbf{h}(x)$  rather than  $\mathbf{h}(x)$  in the programme so that the programme will have a corresponding Dung network. To make  $\neg \mathbf{h}(x)$  fail, we need to add  $\mathbf{h}(x)$ , for  $x = \text{usable}$ , under  $h$ .

The corresponding Dung network for the above network is Figure 10 (Footnote 1):

So, the BG programme is defined to contain the following clauses

- (1\*) If  $x_1, \dots, x_n$  attack  $y$  in the network, we include the clause

$$y \text{ if } \neg x_1 \wedge \dots \wedge \neg x_n \wedge \neg \mathbf{h}(y),$$

where  $\mathbf{h}(y)$  is a new atom.

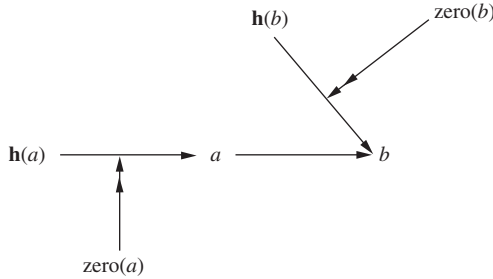


Figure 10. The network of Remark 3.9.

- (2\*) If  $y$  is not attacked by any node, we include the clause  $y$  if  $\neg \mathbf{h}(y)$ .
- (3\*) Add  $\mathbf{h}(y)$  if  $\neg \mathbf{zero}(y)$ .
- (4\*) For every  $y$  such that  $h(y) = 1$ , we include the literal  $\mathbf{zero}(y)$ .<sup>4</sup>

#### 4. Kripke models for argumentation networks 1

We begin this section with some methodological remarks and general examples which will bring us to a point of view best suited for the presentation of modal and temporal argumentation networks.

We begin with a simple example. Consider the sentence

- John read *le livre* with interest.

This sentence contains some French words. To understand the sentence, we need to go to a French dictionary and come back with values. We come back with the words ‘the book’.

Now consider

- John is dishonest because he did not pay yesterday.

To check the value of the above, we must go to yesterday and verify that John did not pay.

Let  $t$  label the location of the main sentence and let  $s$  label the location of where we need to go. In each case, we have the following situation.

In the process of evaluating the algorithm  $\mathcal{A}_t$  at  $t$ , we hit upon a unit of the form

*Take  $x$  to location  $s$ , find a value  $y = V_s^t(x)$  for  $x$  at  $s$  and come back and plug it into our local algorithm  $\mathcal{A}_t$  and carry on.*

The notation  $V_s^t(x)$  is the value you get for  $x$  at location  $s$  intended to be understood and used at  $t$  (so  $V_s^t$  is a French to English ‘function’ in the first example, and a function reading from a payment ledger for the second example).

Let us now take another example. Consider the three argumentation networks in Figure 11. In network  $t$ , the node  $x$  is not an argument but an instruction to look for an accessible network in which there is a winning argument which can defeat  $b$ . The two accessible networks are  $s$  and  $r$ . In  $s$ ,  $a$  is not a winning argument, but it is in  $r$ . Suppose  $a$  is capable of defeating  $b$ ; this knowledge is not recorded in the network  $t$  but is known to us either extra-logically or intrinsically (for example,  $b$  is logically inconsistent with  $a$ ). Then,  $x$  will be instantiated as the argument  $a$  coming from network  $s$ .  $x$ , being an instruction or a recipe for finding arguments, can be as specific as needed for a successful search. Note that we could have written Figure 12 instead of Figure 11. This is a fibring of network  $r$  at position  $x$  at network  $t$ . The attack on  $b$  comes from inside the network  $r$  from node  $a$  onto node  $b$  (in network  $t$ ).

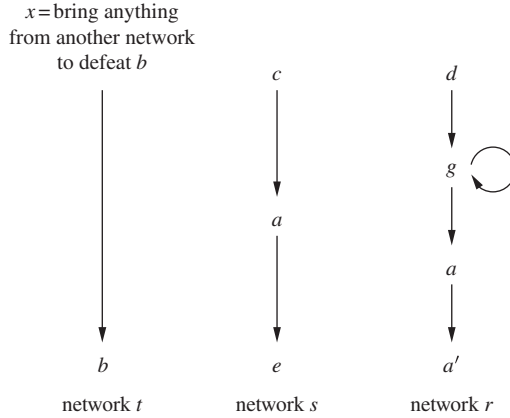


Figure 11. Networks related by instructions.

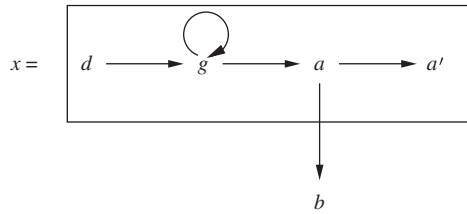


Figure 12. Traditional network corresponding to Figure 11.

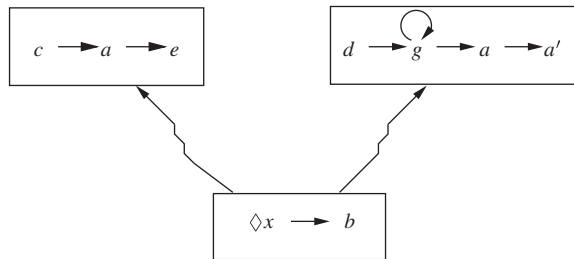


Figure 13. A Kripke model corresponding to Figure 11.

We can turn the situation into modal logic by using  $\diamond x$  (or even  $\diamond a$  if we know that  $x = a$  will do the job) and we get a kind of Kripke model, see, for example, Figure 13. The zigzag arrow  $\rightsquigarrow$  is accessibility and the ordinary arrow  $\rightarrow$  is attack.  $\diamond x$  (or  $\diamond a$ ) means find a winning argument  $a$  which can attack  $b$ . Here, the meaning of  $\diamond a$  is administrative.  $\diamond$  is a meta-level administrative connective.  $\diamond a$  does not mean that  $a$  is a possible argument; and  $\diamond$  is not in the argument language.

Consider now the following network

$$\diamond \text{ storm} \rightarrow b$$

in which  $\diamond \text{ storm}$  represents ‘it is possible there will be a storm’ and  $b$  represents ‘setting sail now’. The model envisages several possible futures, if in at least one of them there is a storm then that possibility defeats  $b$ . Here, the possible  $\diamond$  is not an administrative but a temporal event. The language of  $\diamond a$  is object level, and  $\diamond$  must be in the language of arguments.

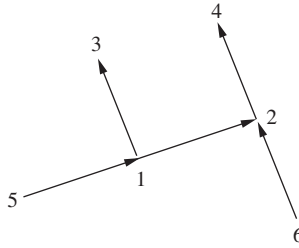


Figure 14. A Kripke model corresponding to Figure 5.

Technically, the mathematics of both cases, the administrative meta-level  $\diamond$  and the temporal event object level  $\diamond$ , is very similar. In both cases, we can put  $\diamond x$  in the nodes of an argumentation network and seek winning arguments in accessible networks.

In either case of  $\diamond x$ , we need to search other networks for an appropriate winning value. So, it is not clear until after the calculation and search, whether we have a usable argument here or not, especially if the family of networks is complex. Hence, the need and the technical usefulness and value of a usability assignment. It simplifies matters during the calculations.

We now explore our options for Kripke models for argumentation networks. We begin with a simple first attempt which will turn out to need improvement. However, it is helpful to go through this first attempt for us to appreciate what is to be added. Consider the network of Figure 5 and let Figure 14 describe a Kripke model. The reader should bear this in mind when reading the next formal definition.

DEFINITION 4.1 (Languages) We recall here the languages of modal and temporal logics.

- (1) The classical connectives we use are  $\sim$  (negation),  $\wedge$ ,  $\vee$ ,  $\rightarrow$ . We reserve  $\neg$  for negation as failure.
- (2) In temporal logic, we use the connective  $PA$  which for  $A$  was true in the past and  $FA$  for  $A$  will be true in the future. A temporal model is usually presented through a nonempty set  $T$  of moments of time and an earlier-later relation  $<$  on  $T$ .  $<$  is usually taken as irreflexive and transitive. The classical truth conditions for  $P$  and  $F$  are
  - $t \models PA$  iff for some  $s$  such that  $s < t$ , we have  $s \models A$ .
  - $t \models FA$  iff for some  $s$  such that  $t < s$ , we have  $s \models A$ .

Temporal logic defines

$HA = \sim P \sim A = A$  has always been true.

$GA = \sim F \sim A = A$  will always be true.

- (3) Modal logic uses  $\diamond A$  reading,  $A$  holds in another accessible world. The set of worlds is denoted by  $S$  and has an accessibility relation  $R \subseteq S \times S$ . We have
  - $t \models \diamond A$  iff for some  $s$  such that  $tRs$  we have  $s \models A$ .

$\Box A$  usually means  $\sim \diamond \sim A$ .
- (4) The usual temporal or modal logics have formulas evaluated at worlds. If we want to define the notion of modal and temporal networks, we will need to deal with networks of formulas evaluated at worlds.
- (5) We can have in the language both the temporal connectives  $P, F$  and the modal connective  $\diamond$ . In which case, the semantics will need to have both  $R$  and  $<$ . We may allow for the future to be also a possibility for  $\diamond$ , in which case we have:

$$t < s \rightarrow tRs.$$

Sometimes only  $P$  and  $\diamond$  are used in which case we can use only  $R$  and go backwards in it for evaluating  $P$ .

- (6) The examples below use ‘usability’ instead of truth. So, we have for an example that ‘ $a$  is usable at world  $t$ ’ is treated mathematically the same way as we treat ‘ $a$  is true (holds) at  $t$ ’.

Let us start with some examples in the simple language which contain arguments of the form  $x$ , (atomic),  $\diamond x$ , possibility of an argument, and  $Px$ , past arguments. Think of  $\diamond x$  as future possibility. So, in the Kripke model,  $\diamond$  goes up in the arrow direction and  $P$  goes down in the arrow direction.

In the usual Kripke evaluation procedures for classical logic, with a set  $Q$  of atomic arguments, we have an assignment  $h$  giving a truth value at each world  $t$  for each atomic proposition  $q$ . We write  $h(t) \subseteq Q$  for the set of true propositions at  $t$ . In the argumentation case, we do not have atomic propositions, but we do have argumentation networks which themselves contain atomic propositions. For example, the network of Figure 5 contains the atoms  $a, b, c$ .

So, we give the following definition. For each node  $n$  in the Kripke model, we assign a set  $h(n) \subseteq$  set of atoms appearing in the network. For an atom  $q$ , we assign a usability value 1 if  $q \in h(n)$  and 0 otherwise. We also use the notation  $h(n, q) = 1$  or  $h(n) \models +q$  to indicate that  $q \in h(n)$ , and  $h(n, q) = 0$  or  $h(n) \models -q$  to indicate that  $q \notin h(n)$ .

Figure 15 is an example of such an assignment, where we write  $\pm q$  to indicate the value of  $q$ .

Suppose we want to know the value of the network of Figure 5 at the model at node 1. How do we evaluate  $Pc$ ? We follow the traditional steps of evaluation in a Kripke model; we go down the accessibility relation and look for a world where  $c$  is usable.

At node 5, we have  $+c$ , so maybe we say  $+Pc$  at node 1, but we notice that  $a$  attacks  $Pc$  in the network (Figure 5). So, does  $1 \models Pc$  or not? Furthermore,  $Pc$  attacks  $b$  and so at node 2 does  $Pb$  hold or not?

We have  $+b$  at node 1, so we would like to say  $2 \models Pb$ , however  $b$  may be successfully attacked at node 1. So, we may not have  $Pb$  after all, so what do we have?

We need an agreed recursive definition.

Let us see some examples where we might have a loop, and try and get a clue by working the example out.

*Example 4.2* Consider the network and Kripke model as described in Figure 16

To evaluate the network at 1, we know that  $\sim Pb = usable$ . But  $\sim Pb$  is attacked by  $\diamond a$  and so we need the value of  $\diamond a$  at 1. For this, we need the value of  $a$  at point 2. We have  $+a$  at 2, but this is attacked by  $\sim Pb$ , which is not attacked by  $\diamond a$  at 2 since it is not usable at 2.

So, we need to know the value of  $b$  at 1. We do have  $+b$  at 1 but this is attacked by  $\diamond a$  so we need to know the value of  $a$  at 2 and we have a loop.

We need some process of evaluation which will give us a better chance to resolve the loops.

We do this by levels of recursive evaluation. We use two bits of notation.

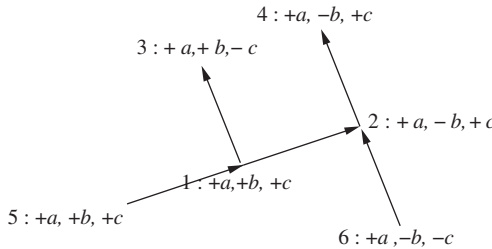


Figure 15. An example of an assignment.



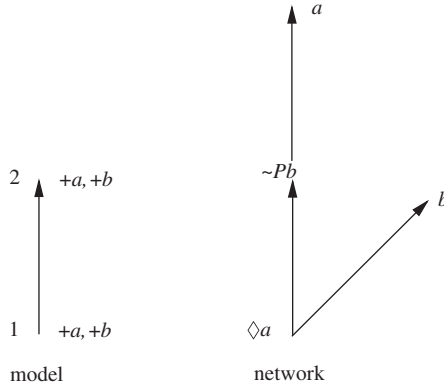


Figure 16. A model for Example 4.2.

- (1)  $h_m(t, A)$  is the assignment of value at level  $m$  to the argument  $A$ .  $A$  may be atomic or a formula.
- (2)  $t \models_m A$  is the network value of  $A$  at world  $t$  at level  $m$ . We use the BG algorithm for the functional  $\pi^*$  namely clauses (1\*)–(3\*) of Remark 3.8.

So to make the meaning of  $h_m$  and  $\models_m$  crystal clear:  $h_m$  says which arguments in the network are usable at level  $m$ , while  $\models_m$  says which arguments are winning (not defeated) at level  $m$ . An argument defeated at  $\models_m$  is considered unusable by  $h_{m+1}$ .

Let us now apply this to Figure 16's example.

*Level 0*

$h_0$  is the assignment  $h$  to the atoms as indicated in the Figure, i.e.  $h_0(1, a) = h_0(1, b) = h_0(2, a) = h_0(2, b) = 1$ .  $\models_0$  is defined for  $ab$  as the same value as  $h_0$ .

For  $\diamond a, Pb, h_0, \models_0$  is not necessarily defined.

It is convenient to use the notation

$$h_m(t) \models +A \text{ to say } h_m(t, A) = 1,$$

$$h_m(t) \models -A \text{ to say } h_m(t, A) = 0.$$

The idea of  $h_m$  and  $\models_m$  is as follows:  $h_m$  evaluates the temporal modalities using the Kripke model without regard to the argumentation network. Then,  $\models_m$  records the result of the attacks (i.e. the winning arguments) of the argumentation network at each world. The  $\models_m$  may record the defeat of some atoms in the network, thus rendering them unusable at level  $m + 1$  (by virtue of being defeated at level  $m$ ) thus giving rise to a new assignment which can now be used to calculate  $h_{m+1}$  and so on.

*Level 1*

$$h_1(1) \models +\diamond a, +b, + \sim Pb, +a,$$

$$h_1(2) \models -\diamond a, +b, - \sim Pb, +a.$$

Now we have networks with nodes which have values and so we calculate the winning arguments. These are the ones holding at the worlds of the Kripke model at level one. We write:

$$1 \models_1 \diamond a, a,$$

$$2 \models_2 b, a.$$

*Level 2*

The assignment we use to calculate  $h_2$  is the atomic part of  $\models_1$ , namely  $1 \models_1 a$  and  $2 \models_1 a, b$ .

$$h_2(1) \models +\diamond a, -b, +a, + \sim Pb,$$

$$h_2(2) : -\diamond a, +b, + \sim Pb, +a.$$

After evaluation of the networks, we get

$$1 \models_2 \diamond a, a,$$

$$2 : \models_2 \sim Pb, b.$$

*Level 3*

$\models_2$  gives us a new assignment to the atoms, namely  $1 \models_2 a$  and  $2 \models_2 b$ .

$$h_3(1) \models -\diamond a, -b, +a, + \sim Pb,$$

$$h_3(2) \models -\diamond a, +b, + \sim Pb, -a.$$

calculating the surviving arguments at each world, we get

$$1 \models_3 \sim Pb, b,$$

$$2 \models_3 \sim Pb, b.$$

*Level 4*

We now get the assignment from  $\models_3$  for atoms as

$$1 \models_3 b, 2 \models_3 b.$$

We calculate  $h_4$

$$h_4(1) \models -\diamond a, - \sim Pb, +b, -a,$$

$$h_4(2) \models -\diamond a, - \sim Pb, +b, -a.$$

We calculate  $\models_4$  at each node using network rules

$$1 \models_4 \sim Pb, b,$$

$$2 \models_4 \sim Pb, b.$$

$\models_3$  and  $\models_4$  are the same. So, the answer is now stable.

*Remark 4.3* The reader may wonder what has happened to the loop we observed before and what kind of interpretation (loop resolution) we are getting. To explain that let us look at the traditional loop in the network of Figure 17, in which  $a$  and  $b$  attack each other. We have three complete extensions  $\{a\}, \{b\}, \emptyset$ .<sup>5</sup> Let us do the level calculation intuitively.

*Level 0*

Start with  $+a, +b$ .

*Level 1*

Attack as suggested by level 0. We get  $-a, -b$ .

*Level 2*

Attack as suggested by level 2. We get  $+a, +b$ .

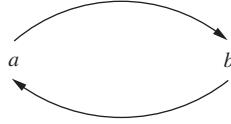


Figure 17. A traditional loop for Remark 4.3.

So, we are infinitely looping and we can put a question mark on  $a$  and  $b$ , leading to  $\emptyset$ .

Of course, everything depends on the assignment at level 0. Other possible assignments are  $\{+a, -b\}$ ,  $\{-a, +b\}$ ,  $\{-a, -b\}$ , yielding  $\{a\}$ ,  $\{b\}$ , and  $\emptyset$ , respectively.

*Example 4.4* We try and evaluate the network of Figure 5 at the model of Figure 16. We use the BG  $\pi^*$  evaluation algorithm.

We do this by levels. Let  $h_0$  be the assignment to the atoms as indicated in Figure 15. Thus,  $h_0$  satisfies

*Level 0*

$$h_0(5) \models +a, +b, +c,$$

$$h_0(6) \models +a, -b, -c,$$

$$h_0(1) \models +a, +b, +c,$$

$$h_0(2) \models +a, -b, -c,$$

$$h_0(3) \models +a, +b, -c,$$

$$h_0(4) \models +a, -b, +c.$$

*Level 1*

$$h_1(5) \models +a, +b, +c, -Pc, +\diamond a,$$

$$h_1(6) \models +a, -b, -c, -Pc, +\diamond a,$$

$$h_1(1) \models +a, +b, +c, +Pc, +\diamond a,$$

$$h_1(2) \models +a, -b, +c, -Pc, +\diamond a,$$

$$h_1(3) \models +a, +b, -c, +Pc, -\diamond a,$$

$$h_1(4) \models +a, -b, +c, +Pc, -\diamond a.$$

Now the network of Figure 5 has values for each node at each world. We can compute the winning argument at each world. Note that  $c$  does not appear as a node in the network so we inherit its value from  $h$ .

$$5 \models_1 +a, -b, +c, -Pc, +\diamond a,$$

$$6 \models_1 +a, -b, -c, -Pc, +\diamond a,$$

$$1 \models_1 +a, -b, +c, +Pc, +\diamond a,$$

$$2 \models_1 +a, -b, +c, -Pc, +\diamond a,$$

$$3 \models_1 +a, +b, -c, +Pc, -\diamond a,$$

$$4 \models_1 +a, +b, +c, +Pc, -\diamond a.$$

*Level 2*

We evaluate  $h_2$  using the assignment to the atoms suggested by  $\models_1$ .

$$\begin{aligned}
h_2(5) &\models +a, -b, +c, -Pc, +\diamond a, \\
h_2(6) &\models +a, -b, +-, -Pc, +\diamond a, \\
h_2(1) &\models +a, -b, +c, +Pc, +\diamond a, \\
h_2(2) &\models +a, -b, +c, -Pc, +\diamond a, \\
h_2(3) &\models +a, +b, -c, +Pc, -\diamond a, \\
h_2(4) &\models +a, +b, +c, +Pc, -\diamond a.
\end{aligned}$$

We now calculate  $\models_2$

$$\begin{aligned}
5 &\models_2 +a, -b, +c, -Pc, +\diamond a, \\
6 &\models_2 +a, -b, -c, -Pc, +\diamond a, \\
1 &\models_2 +a, -b, +c, +Pc, +\diamond a, \\
2 &\models_2 +a, -b, +c, -Pc, +\diamond a, \\
3 &\models_2 +a, +b, -c, +Pc, -\diamond a, \\
4 &\models_2 +a, +b, +c, +Pc, -\diamond a.
\end{aligned}$$

We have stability since  $\models_1$  equals  $\models_2$ .

**DEFINITION 4.5 (Temporal languages)** Let  $Q$  be a set of atoms. The basic temporal language based on  $Q$  uses the unary connectives  $\mathbb{C} = \{\neg, \diamond, P\}$ . A temporal formula has the form  $\alpha_1\alpha_2\dots\alpha_mq$ , where  $q \in Q$  and  $\alpha_i \in \mathbb{C}, i = 1, \dots, m$ .

The language is said to be very basic if we allow  $\diamond x$  and  $Px$  and  $x$  only.

The full temporal language also allows the use of the classical connectives  $\wedge, \vee \rightarrow$  and unrestricted use of  $\diamond$  and  $P$ .

**DEFINITION 4.6 (Temporal Kripke models)** Let  $(S, R, a)$  be a Kripke model and let  $\mathbb{N} = (\mathbb{F}, \rho)$  be a network in the basic temporal language with  $\neg, \diamond, P$ , and  $Q$  or in the full temporal language. Let  $h$  be an assignment giving for each world  $t \in S$  and an atom  $q \in Q$  a usability value  $h(t, q) \in \{0, 1\}$ .<sup>6</sup>

We define by induction a sequence of new assignments  $h_1, h_2, \dots$  and semantic consequences  $\models_1, \models_2, \dots$  as follows:

- (1) Let  $h_1$  be defined as follows

$$h_1(t, q) = h(t, q), \text{ for } q \text{ atomic,}$$

$$h_1(t, \diamond A) = 1 \text{ if for some } s \in S \text{ such that } tRs \text{ we have } h_1(s, A) = 1.$$

$$h_1(t, PA) = 1 \text{ iff for some } s, \text{ such that } sRt \text{ we have } h_1(s, A) = 1.$$

The definition for the classical connectives is the usual one.

- (2) Let  $\models_1$  be defined as follows:

First consider  $h_1(t)$  as an assignment on  $(\mathbb{F}, \rho)$  with  $t$  as a fixed parameter. We can consider  $h_1(t)$  as a subset of  $\mathbb{F}$ . Consider the operator  $\pi$  of Definition 3.3.

We define  $\models_1$  by

$$t \models_1 x \text{ iff } x \in \pi h_1(t).$$

The reader can compare with the construction in Example 4.2.

Note that if we want to use defeasible rules as discussed in Remark 3.8, then we use  $\pi^*$  of Remark 3.8 instead of  $\pi$  of Definition 3.3.

We now define  $h_{m+1} \models_{m+1}$  for  $m \geq 1$ .

$h_{m+1}$  is obtained from  $\models_m$  in the same way that  $h_1$  was obtained from  $h$ , by regarding  $t \models_m q, q \in Q$  as an assignment to the atoms. Note that all we need is the values of  $\models_m$  on the atoms of  $Q$ .

$\models_{m+1}$  is obtained from  $h_{m+1}$  in the same way that  $h_2$  was obtained from  $h_1$ , i.e. for each  $t$ , we have  $t \models_{m+1} A$  iff  $A \in \pi h_{m+1}(t)$ .

This defines  $h_m \models_m$  for all  $n \geq 1$ .

We now define  $t \models_\infty A$  for  $t \in S$  and  $A \in \mathbb{F}$  as follows:

$t \models_\infty A$  holds (does not hold) if for some  $k$ ,  $t \models_m A$  holds (resp. does not hold) for all  $n \geq k$ .

Otherwise if  $t \models_m A$  oscillates, we say that  $t \models_\infty A$  is undecided.

### 5. Kripke models for argumentation networks 2

Let us assess the situation we are in. In the previous section, we offered a simple model. This model can be improved. The problem is not so much the discrepancy with the Dung approach (Remark 3.9) as this is not a unique possible world problem, but the difficulty is that we need to sharpen the intuitive meaning to Definition 4.6. It is mainly technically motivated by a natural formal analogue of the semantical movements in a traditional modal and temporal Kripke model. We need to clarify more sharply the meaning of usability of arguments and its connection to truth and falsity and possibility of argument and facts.

There seems to be a fundamental difference between the modal operator  $\diamond$  and the temporal operator  $P$ .  $P$  goes into the past, while  $\diamond$  goes to an alternative world of different reasoning/argumentation framework. In an ordinary Kripke semantics for traditional modal and temporal logics, there is no technical distinction between the two. In an argumentation context, we need to give different technical treatment to these two connectives. The  $\diamond$  we treat as a fibring operator (go to another context, do something there, and come back with the result, see Gabbay (1996)), while the operator  $P$  is still treated as purely temporal.

We illustrate with an example.

We want to argue against a political candidate  $c$ . We want to bring in the past facts that he double-crossed his partners, showing lack of loyalty and trustworthiness (call this  $Pd$ ,  $d$  for ‘dirt’). However, the situation today is such that digging up the past on a candidate is counterproductive (call this  $\sim p$ ). It is suggested therefore to wait 6 months for the facts to emerge naturally (i.e.  $\diamond d$ , where  $\diamond$  here reads future possibility).

A counterargument against waiting is that by that time criteria for judging candidates will change and the argument will be defeated, say  $d$  will be attacked by  $e$  ( $e$  can mean who cares? ; it was a long time ago!).

Figure 18 shows the situation:

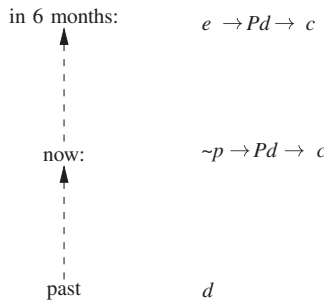


Figure 18. A network with a temporal argument.

We notice the following discrepancies between the formal situation of Figure 18 and the formal definition we gave to modal and temporal argumentation networks in Definition 4.6.

- (1) We have different networks at different possible worlds.
- (2)  $\diamond$  behaves like a fibring operator
- (3)  $P$  is purely temporal for facts.
- (4) With  $P$  the assignment,  $h$  indicates  $\pm$  usability by virtue of truth or falsity, while with  $\diamond$ , the assignment  $h$  might indicate  $\pm$  usability for other reasons.
- (5) We cannot have a proper temporal and modal treatment of arguments without looking into the details of what is the internal structure of arguments and how exactly do they attack one another in terms of such structure.

Suppose we adopt the view that arguments are proofs and attacks disrupt such proofs. Let us examine how time  $T$  gets into the picture. Consider the following example:

I park my car near Russell Square at 11am in the morning, do my business of the day and come back to it at 5 pm. I expect to find it there. If I don't find the car then nonmonotonic deduction allows me to conclude that the car was stolen. We can have a little logical system (nonmonotonic or Bayesian net) which compares the conclusions of 'car stolen' against 'car towed away by local council parking department'. We can assume the latter conclusion can be defeated. We can also ignore some well-known difficulties of persistence of arguments where one gets paradoxically that the car was stolen only a few seconds before my return (the stolen car paradox).

There is one sure way to attack this argument and its conclusion. This is to prove the simple fact that I do not have a car.

Figure 19 illustrates the situation

Facts can attack arguments most effectively. Also, if the facts are undecided or are not available, then we claim the argument is not usable. See Section 6 for further discussion. So when designing a new modal and temporal logic for argumentation, we need a pure  $\{P, F\}$  temporal logic just for the facts.

- (6) Our next question is whether an argument itself can be time dependent. This is a bit tricky. In monotonic logic, the answer is no. Euclid geometric proofs are as valid and good today as they were in ancient times. But in the nonmonotonic case, the answer is yes. Nonmonotonic reasoning depends on context. Today, a girl in a mini-skirt will not be considered immodest but go back 200 years and everyone at that time will nonmonotonically deduce she is 'fast'. So as we can see from this example, the deduction mechanism itself can change in time. Thus, we may argue for example along the lines 'you had better get yourself a decent long dress now because soon people's perception will change and you will no longer be respectable wearing a mini-skirt'.

DEFINITION 5.1 (Temporal Kripke models – updated)

- (1) Consider a language with the classical connectives  $\{\sim, \wedge, \vee, \rightarrow\}$  and the temporal and modal connectives  $\{P, F, \diamond\}$ . We also assume a set  $Q$  of atoms, and we use the atoms to construct formulas using the connectives in the traditional manner.

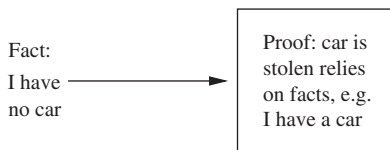


Figure 19. Facts attacking an argument.

- (2) A Kripke model for the above language has the form  $(S, R, \leq, a, h)$ , where  $S$  is a nonempty set of worlds and  $R \subseteq S \times S$  is a binary relation for  $\diamond$  and  $<$  is an irreflexive and transitive relation on  $S$  for  $F$  and  $P$ .  $h$  is an assignment giving for each  $t \in S$ , a subset  $h(t) \subseteq Q$ . We consider  $h$  as a usability function on the atomic arguments of  $Q$ , saying which elements of  $Q$  are usable at world  $t$ .
- (3) For each  $t$ , let  $\mathbb{N}(t) = (\mathbb{F}(t), \rho(t))$  be an argumentation system.
- (4) We define by induction a sequence of new assignments  $h_1, h_2, \dots$  and semantic consequences  $\vDash_1, \vDash_2, \dots$  as follows:
- (a) Let  $h_1$  be defined as follows
- $h_1(t, q) = h(t, q)$ , for  $q$  atomic.
  - $h_1(t, \diamond A) = 1$  if for some  $s \in S$  such that  $tRs$  we have  $h_1(s, A) = 1$ .
  - $h_1(t, PA) = 1$  iff for some  $s$ , such that  $s < t$  we have  $h_1(s, A) = 1$ .
  - $h_1(t, FA)$  iff for some  $s$  such that  $t < s$  we have  $h_1(s, A) = 1$ .
- The definition for the classical connectives is the usual one.
- (b) Let  $\vDash_1$  be defined as follows:
- First consider  $h_1(t)$  as an assignment on  $(\mathbb{F}, (t), \rho(t))$  with  $t$  as a fixed parameter. We can consider  $h_1(t)$  as a subset of  $\mathbb{F}(t)$ . Consider the operator  $\pi^*$  of Remark 3.8. We define  $\vDash_1$  by

$$t \vDash_1 x \text{ iff } x \in \pi^* h_1(t).$$

We now define  $h_{m+1} \vDash_{m+1}$  for  $m \geq 1$ .

$h_{m+1}$  is obtained from  $\vDash_m$  in the same way that  $h_1$  was obtained from  $h$ , by regarding  $t \vDash_m q, q \in Q$  as an assignment to the atoms. Note that all we need is the values of  $\vDash_m$  on the atoms of  $Q$ .

$\vDash_{m+1}$  is obtained from  $h_{m+1}$  in the same way that  $h_2$  was obtained from  $h_1$ , i.e. for each  $t$ , we have  $t \vDash_{m+1} A$  iff  $A \in \pi^* h_{m+1}(t)$ .

This defines  $h_m \vDash_m$  for all  $n \geq 1$ .

We now define  $t \vDash_\infty A$  for  $t \in S$  and  $A \in \mathbb{F}$  as follows:

$t \vDash_\infty A$  holds (does not hold) if for some  $k$ ,  $t \vDash_m A$  holds (resp. does not hold) for all  $n \geq k$ . Otherwise if  $t \vDash_m A$  oscillates, we say that  $t \vDash_\infty A$  is undecided.

## 6. Conclusion and discussion

Modal logic deals with sets  $W$  of possible worlds  $w \in W$ . The possible worlds in propositional modal logics are usually atomic and have no internal structure. In this paper, we associate with possible worlds  $w$  argumentation networks of the form  $N_w$ . The argumentation networks  $N_w$  contain as arguments (i.e. the arguments appearing in the argumentation network  $N_w$ ) modal and temporal formulas. Thus, to define extension in  $N_w$ , we need to look at extensions in other accessible worlds  $N'_w$ . This greatly enriched the language of argumentation but required a tricky inductive definition of how to calculate extensions and required the concept of usability of arguments. We also gave examples to motivate the need for such definitions, but we also discussed simpler definitions of temporal argumentation networks.

We continue with a comparison with some key papers. I am grateful to the referees for compiling this list.

- (1) M.L. Cobo, D.C. Martinez, and G. Simari, a 2010 paper on admissibility in timed abstract argumentation networks (Cobo, Martinez, and Simari 2010).  
This paper offers a time stamping model, where each argument is time stamped when it is available. The model is similar but not the same as the ones used in Abraham et al.

- (2011a) and Barringer et al. (2005). In all cases, the change is in the meta-level. There are no temporal connectives involved.
- (2) N.D. Rotstein, M.O. Moguillansky, A.J. Garcia, and G.R. Simari, a 2010 paper on dynamic argumentation framework (Rotstein, Moguillansky, Garcia, and Ricardo 2010).  
The paper offers a model where arguments are based on evidence structures. Thus, the dynamics of change come from updates and changes in the evidence structure. There are no modal connectives in the object level to connect between different evidence structures.
  - (3) G. Boella, S. Kaci, and L. van der Torre, a 2009 paper on dynamics in argumentation with single extensions (Boella, Kaci, van der Torre 2009).  
This paper does not offer dynamics of argumentation networks but rather principles in the meta-level which can govern change in networks. In particular, they offer properties for refinement of the attack relation. Thus, if, for example, we have an interrelated family of argumentation networks (different worlds or different times), we can check whether this family satisfies the meta-level principles offered in this paper.

Let us conclude with our plans for future research. What is interesting is to look at an argumentation network and partition the arguments into different subsets and regard them as subnetworks (i.e. different modal worlds) and put requirements on extensions as seen from the point of view of the different subnetworks. This presents a very neat way of introducing modality into argumentation.

### Acknowledgements

We are grateful to the referees for most valuable penetrating comments on this paper and on paper (Barringer et al. 2012). Research completed under the Russia–Israel Scientific Research Cooperation, in the project: Combined Modal Logics.

### Notes

1. As an argument for wishing a new interview date, the candidate has declared that she is getting married in the local catholic church, and these dates coincide. As a result of this correspondence, the interviewers know she is Catholic and a new bride.
2. There is a known case where a preferred candidate did not score as well as another candidate in an interview for a position in local government. In exceptional circumstances, the interview panel was reconvened and the outcome was that the preferred candidate's score actually fell!
3. In terms of Definition 3.1,  $H$  is a labelling  $\mathbf{I}$  (no transmission labels) and  $\pi$  is a functional  $\mathbf{f}$  whose algorithm uses clauses (1)–(3).
4. If you recall the idea of attacks on attacks as developed in Section 2 of Barringer, Gabbay, and Woods (2012) and originally in Barringer et al. (2005), you will realise that we can present a network relative to additional nodes attacking connections. The  $\mathbf{zero}(y)$  nodes are added according to assignment to attack the connection  $\mathbf{zero}(y) \rightarrow \mathbf{h}(y)$ . In fact, there is no need to do it this way. We do not need  $\mathbf{zero}(y)$ . We can simply augment the original network with additional nodes  $\mathbf{h}(y)$  attacking the node  $y$  for all  $y$  such that  $h(y) = 0$ . The problem with that is that  $h$  keeps on changing and so the  $\mathbf{h}(y)$  will keep on being in and out.
5. (1) A subset  $E$  of arguments  $E$  is conflict free if no member of it is attacked by another member.  
(2) A subset of arguments  $E$  is admissible if whenever a member  $x$  in  $E$  is attacked by any other argument  $z$ , then there is a  $y$  in  $E$  which attacks  $z$ .  
(3)  $E$  is preferred extension if it is a subset satisfying (1) and (2) and is maximal with respect to set inclusion.
6. This definition is parallel to the traditional one.  $\diamond$  goes up the accessibility relation and  $P$  goes down it. The evaluation is more complicated because the formulas are part of a network.



As noted before if we insist that at each  $t$ , the assignment  $h(t) \subseteq Q$  is a Caminada assignment (need not be the same one for all  $t$ ), then we will have less technical discrepancies with the Dung interpretation. We need to check, however, since our language has temporal operators, whether  $h_m(t)$  remains a Caminada assignment. We will examine this point later in the section. Our guess is that further adjustment will be needed.

We shall give better definitions later in the next section.

## References

- Abraham, M., Gabbay, D.M., and Schild, U. (2011a), ‘The Handling of Loops in Talmudic Logic, with Application to Odd and Even Loops in Argumentation’, in *Proceedings of Howard 60*, eds. D. Rydeheard, A. Voronkov, and M. Korovina, pp. 1–25, earlier version published as an appendix in h (Abraham, Gabbay, and Schild 2011b).
- Abraham, M., Gabbay, D.M., and Schild, U. (2011b), *Resolution of Conflicts and Normative Loops in the Talmud*, College Publications, London, UK.
- Barringer, H., Gabbay, D.M., and Woods, J. (2005), ‘Temporal Dynamics of Support and Attack Networks’, in *Mechanising Mathematical Reasoning*, eds. D. Hutter and W. Stephan, LNCS Vol. 2605, Springer, Springer-Verlag Berlin, Heidelberg, pp. 59–98.
- Barringer, H., and Gabbay, D.M. (2010), ‘Modal and Temporal Argumentation Networks’, in *Amir Pnueli Memorial Volume: Time for Verification*, eds. D. Peled and Z. Manna, LNCS Vol. 6200, Springer, Springer-Verlag Berlin, Heidelberg, pp. 1–25.
- Barringer, H., Gabbay, D.M., and Woods, J. (2010), ‘Temporal, Numerical and Meta-level Dynamics in Argumentation Networks’, to appear in *Argumentation and Computation*, Special issue, 3(2–3), 2012.
- Boella, G., Kaci, S., and van der Torre, L. (2009), ‘Dynamics in Argumentation with Single Extensions: Attack Refinement and the Grounded Extension (Extended Version)’, *Proceeding ArgMAS’09 Proceedings of the 6th international conference on Argumentation in Multi-Agent Systems*, Springer-Verlag Berlin, Heidelberg ©2010, pp. 150–159.
- Caminada, M. (2006), ‘On the Issue of Reinstatement in Argumentation’, *Jelia*, pp. 111–123. [http://icr.uni.lu/martinc/publications/JELIA\\_reinstatement.pdf](http://icr.uni.lu/martinc/publications/JELIA_reinstatement.pdf)
- Caminada, M., and Gabbay, D.M. (2009), ‘A Logical Account of Formal Argumentation’, *Studia Logica*, 93(2–3), 109–145.
- Caminada, M. (2011), ‘A Labelling Approach for Ideal and Stage Semantics’, *Argument and Computation*, 2(1), 1–21.
- Cobo, M.L., Martinez, D.C., and Simari, G.R. (2010), ‘On Admissibility in Timed Abstract Argumentation Frameworks’, Published in: *Proceeding Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, IOS Press Amsterdam, The Netherlands, The Netherlands ©2010, pp. 1007–1008.
- Gabbay, D.M. (1996), *Fibring Logics*, OUP, Oxford UK.
- Gabbay, D.M. (2011), ‘Dung’s Argumentation is Equivalent to Classical Propositional Logic with the Peirce-Quine Dagger’, *Logica Universalis*, 5(2), 255–318, DOI : 10.1007/s11787-011-0036-3.
- Jakobovits, H., and Vermeir, D. (1999), ‘Robust Semantics for Argumentation Frameworks’, *Journal of Logic and Computation*, 9, 215–161.
- Rotstein, N.D., Moguillansky, M.O., Garcia, A.J., and Ricardo, G. ‘Simari: A Dynamic Argumentation Framework’, Published in: *Proceeding Proceedings of the 2010 conference on Computational Models of Argument: Proceedings of COMMA 2010*, IOS Press Amsterdam, The Netherlands, The Netherlands ©2010, pp. 427–438.