

# The Attack as Intuitionistic Negation

D. Gabbay  
King's College London,  
Department of Informatics,  
Strand, London, WC2R 2LS, UK;  
Bar Ilan University, Ramat Gan, Israel  
and  
University of Luxembourg, Luxembourg.  
dov.gabbay@kcl.ac.uk

M. Gabbay  
Cambridge University, UK.  
mg639@cam.ac.uk

Compiled on October 2, 2015

## Abstract

We translate the argumentation networks  $\mathcal{A} = (S, R)$  into a theory  $\Delta_{\mathcal{A}}$  of intuitionistic logic, retaining  $S$  as the domain and using intuitionistic negation to model the attack  $R$  in  $\mathcal{A}$ : the attack  $xRy$  is translated to  $x \rightarrow \neg y$ . The intuitionistic models of  $\Delta_{\mathcal{A}}$  characterise the complete extensions of  $\mathcal{A}$ .

The reduction of argumentation networks to intuitionistic logic yields, in addition to a representation theorem, some additional benefits: it allows us to give semantics to higher level attacks, where an attack “ $xRy$ ” can itself attack another attack “ $uRv$ ”; one can make higher level meta-statements  $W$  on  $(S, R)$  and such meta-statements can attack and be attacked in the domain.

## 1 Background and orientation

This paper is a continuation of [1] but it is self-contained and is independent of [1], except that it expands the ideas of [1].

Given a finite abstract argumentation network  $\mathcal{A} = (S, R)$ , where  $S \neq \emptyset$  is the set of arguments and  $R \subseteq S \times S$  is the attack relation, we would like to view the set  $S$  as atomic propositions of the intuitionistic propositional calculus and translate the attack relation  $xRy$  as  $x \rightarrow \neg y$ , where “ $\rightarrow$ ” represents intuitionistic implication and “ $\neg$ ” represents intuitionistic negation. For each  $\mathcal{A}$  we write a

{sec1}

theory  $\Delta_{\mathcal{A}}$  such that all the complete extensions of  $\mathcal{A}$  correspond to all the intuitionistic models of  $\Delta_{\mathcal{A}}$ .

The reduction of argumentation networks to intuitionistic logic yields, in addition to a representation theorem, some additional benefits.

- It allows us to give semantics to higher level attacks, where an attack “ $xRy$ ” can itself attack another attack “ $uRv$ ”.
- One can make higher level meta-statements  $W$  on  $(S, R)$  and such meta-statements can attack and be attacked.

For example we can attack an argument  $a$  by saying that  $a$  is a generic argument which attacks all other arguments  $\{x \mid x \neq a\}$  and therefore  $a$  should be out.

What we are saying is  $W(a)$  where:

$$W(a) = \forall x(x \neq a \rightarrow aRx)$$

attacks  $a$ .

We shall use Gödel’s intuitionistic logic  $\mathbf{G}_3$ , semantically defined by all intuitionistic Kripke models with just two linearly ordered worlds  $t < s$  ( $t$  the actual world and  $s$  a possible world), with  $<$  the intuitionistic accessibility relation. Appendix A describes the logic  $\mathbf{G}_3$  in detail. It can be axiomatised.

We present the complete extensions of  $\mathcal{A} = (S, R)$ , using the Caminada labelling approach, [2]. A Caminada labelling of  $S$  is a function  $\lambda : S \mapsto \{\text{in, out, und}\}$  such that the following holds

- (C1)  $\lambda(x) = \text{in}$  iff for all  $y$  attacking  $x$ ,  $\lambda(y) = \text{out}$ .
- (C2)  $\lambda(x) = \text{out}$  iff for some  $y$  attacking  $x$ ,  $\lambda(y) = \text{in}$ .
- (C3)  $\lambda(x) = \text{und}$  iff for all  $y$  attacking  $x$ ,  $\lambda(y) \neq \text{in}$ , and for some  $y_0$  attacking  $x$ ,  $\lambda(y_0) = \text{und}$ .<sup>1</sup>
- (C4) If  $x$  is not attacked at all, then  $\lambda(x) = \text{in}$ .<sup>2</sup>

Let us use the following notation for  $\mathbf{G}_3$ .

For a proposition  $e$ , write  $e = (\top, \top)$  to mean  $t \vDash e$  and  $s \vDash e$ . Write  $e = (\perp, \top)$  to mean  $t \not\vDash e$  and  $s \vDash e$ . Write  $e = (\perp, \perp)$  to mean  $t \not\vDash e$  and  $s \not\vDash e$ .

Note that since  $\{t < s\}$  is an intuitionistic model, the option  $t \vDash e$  and  $s \not\vDash e$  is not allowed.

We denote assignments  $h$  of truth values to atoms in the model by  $h(e) = (\top, \top)$  or  $h(e) = (\perp, \top)$  or  $h(e) = (\perp, \perp)$ . We also write  $e = (\top, \top), (\perp, \top), (\perp, \perp)$  respectively, using abuse of notation, when the assignment is known. So  $t \vDash e$  means  $t \vDash_h e$  and  $s \vDash e$  means  $s \vDash_h e$ .

<sup>1</sup>That is,  $\lambda(x) = \text{und}$  iff neither  $\lambda(x) = \text{in}$  nor  $\lambda(x) = \text{out}$ .

<sup>2</sup>This condition follows from (C1), since the empty conjunction is considered  $\top$ .

## 2 Translation into $\mathbf{G}_3$

{sec2}

Let  $\mathcal{A} = (S, R)$  be a finite argumentation network. Let  $\mathbf{G}_3$  be Gödel's logic with two possible worlds  $t < s$ , with  $t$  the actual world. Let  $S$  be considered as propositional atoms of  $\mathbf{G}_3$ , and let  $\mathbf{n}$  be an additional new constant of  $\mathbf{G}_3$ , whose truth condition is  $(\perp, \top)$  meaning  $t \not\models \mathbf{n}$  and  $s \models \mathbf{n}$ . We thus have that  $t \not\models \mathbf{n} \vee \neg \mathbf{n}$  and  $t \not\models \neg \neg \mathbf{n} \rightarrow \mathbf{n}$ .

We shall use  $\mathbf{n}$  in our translation and eliminate it in the next section.  $\mathbf{n}$  can be characterised by the condition  $\not\models \mathbf{n} \vee \neg \mathbf{n}$ . In other words, we take an axiom system for  $\mathbf{G}_3$  and add to the language the constant  $\mathbf{n}$  with the ‘‘axiom’’ that  $\mathbf{n} \vee \neg \mathbf{n}$  is false.

{541-D1}

**Definition 2.1** *Let  $\mathcal{A} = (S, R)$  and  $\mathbf{n}$  be as described before. Define the theory  $\Delta_{\mathcal{A}}$  containing the following formulas of  $\mathbf{G}_3$ , for  $x, y$  in  $S$ :*

$$(a1) \quad \bigwedge_{x \in S} (x \rightarrow \mathbf{n} \vee \bigwedge_{y R x} \neg y)$$

$$(a2) \quad \bigwedge_{x \in S} (\bigwedge_{y R x} \neg y \rightarrow \mathbf{n} \vee x)$$

$$(b1) \quad \bigwedge_{x \in S} (\neg x \rightarrow \mathbf{n} \vee \bigvee_{y R x} y)$$

$$(b2) \quad \bigwedge_{x \in S} (\bigvee_{y R x} y \rightarrow \neg x \vee \mathbf{n})$$

{541-T2}

### Theorem 2.2

1. *Let  $h$  be a model of  $\Delta_{\mathcal{A}}$ . Define a Caminda labelling  $\lambda_h$  as follows:*

- $\lambda_h(x) = in$ , if  $h(x) = (\top, \top)$
- $\lambda_h(x) = out$ , if  $h(x) = (\perp, \perp)$
- $\lambda_h(x) = und$ , if  $h(x) = (\perp, \top)$ .

*Then  $\lambda_h$  is a complete extension for  $\mathcal{A}$ .*

2. *Let  $\lambda$  be a complete extension of  $\mathcal{A}$ . Define  $h_\lambda(x)$  for  $x \in S$  by*

- $h_\lambda(x) = (\top, \top)$ , if  $\lambda(x) = in$ .
- $h_\lambda(x) = (\perp, \perp)$ , if  $\lambda(x) = out$
- $h_\lambda(x) = (\perp, \top)$ , if  $\lambda(x) = und$ .

*Then  $h_\lambda$  is a model of  $\Delta_{\mathcal{A}}$ .*

### Proof.

**Part 1.** Assume  $h$  is a model of  $\Delta_{\mathcal{A}}$ . We show  $\lambda_h$  satisfies the conditions of Caminda labelling.

**Condition C1.** Assume  $\lambda_h(x) = \text{in}$ . Then  $x = (\top, \top)$ . From axiom (a1) we get, since  $t \models x$  and  $t \not\models \mathbf{n}$  that  $t \models \bigwedge_{yRx} \neg y$  and so  $h(y) = (\perp, \perp)$  for all attackers  $y$  of  $x$ , i.e.  $\lambda_h(y) = \text{out}$  for all attackers of  $x$ .

Assume  $\lambda_h(x) = \text{out}$  for all attackers of  $x$ . Then  $h(\bigwedge_{yRx} \neg y) = (\top, \top)$ . From axiom (a2) we get  $t \models x$  and so  $h(x) = (\top, \top)$  and so  $\lambda_h(x) = \text{in}$ .

**Condition C2.** Assume  $\lambda_h(x) = \text{out}$ . Then  $x = (\perp, \perp)$ . Then from axiom (b1) we have  $t \models \bigvee_{yRx} y$ . Let  $y_0$  be such that  $t \models y_0$ . Then  $y_0 = (\top, \top)$  and so  $\lambda_h(y) = \text{in}$ .

Assume for some  $y_0$  we have  $\lambda_h(y_0) = \text{in}$ . Then  $y_0 = (\top, \top)$  and so from axiom (b2) we get that  $t \models \neg x \vee \mathbf{n}$ . Therefore  $t \models \neg x$  and so  $x = (\perp, \perp)$  and so  $\lambda_h(x) = \text{out}$ .

**Condition C3.** We have  $\lambda_h(x) = \text{und}$  iff by definition  $h(x) = (\perp, \top)$ . From (b2) we have that

$$t \models \left( \bigvee_{yRx} y \rightarrow \neg x \vee \mathbf{n} \right).$$

But at  $t$  we have that  $t \not\models \neg x \vee \mathbf{n}$ . So  $t \not\models \bigvee_{yRx} y$ . So all attackers  $y$  are false at  $t$ . So their value can be either  $(\perp, \perp)$  (i.e. out) or  $(\perp, \top)$  (i.e. undecided) but not  $(\top, \top)$ .

Assume  $x = (\perp, \top)$ . From (a2) we have that

$$t \models \left( \bigwedge_{xRy} \neg y \rightarrow \mathbf{n} \vee x \right).$$

Since  $t \not\models \mathbf{n} \vee x$  we have  $t \not\models \bigwedge_{yRx} \neg y$ . Thus there exists at least one  $y_0$  such that  $t \not\models \neg y_0$ . So either  $y_0 = (\top, \top)$  or  $(\perp, \top)$ . By what we have shown  $y_0$  must be  $(\perp, \top)$ , i.e.  $y_0$  is undecided (und).

Assume now that all attackers of  $x$  are either out or und and one attacker  $y_0$  is und. Thus we have all  $y$  attacking  $x$  are either  $(\perp, \perp)$  or  $(\perp, \top)$  and one  $y_0 = (\perp, \top)$ .

From (b1) we have

$$t \models \neg x \rightarrow \mathbf{n} \vee \bigvee_{yRx} y.$$

Since  $t \not\models \mathbf{n} \vee \bigvee_{yRx} y$  we must have  $t \not\models \neg x$ . So  $x = (\top, \top)$  or  $(\perp, \top)$ .

Finally from (a1) we have

$$t \models x \rightarrow \mathbf{n} \vee \bigwedge_{yRx} \neg y.$$

Since  $y_0 = (\perp, \top)$  we get  $t \not\models \mathbf{n} \vee \bigwedge_{yRx} \neg y$  and so  $t \not\models x$ . Therefore  $x = (\perp, \top)$ , i.e.  $x$  is und.

**Part 2.** Assume  $\lambda$  is a Caminada labelling satisfying (C1), (C2) and (C3). We show that  $h_\lambda$  is a model of  $\Delta_{\mathcal{A}}$ . Since  $s \models \mathbf{n}$ , the axioms clearly hold at  $s$ . Therefore need only show the the axioms of  $\Delta_{\mathcal{A}}$  holds at  $t$ , this is made simpler by the fact that  $t \not\models \mathbf{n}$ .

**Axiom (a1).** Show if  $t \models x$  then  $\models \bigwedge_{yRx} \neg y$ . This follows from condition (C1).  $t \models x$  means  $\lambda(x) = \text{in}$ . So  $\lambda(y) = \text{out}$  for all  $yRx$  so  $t \models \bigwedge_{yRx} \neg y$ .

**Axiom (a2).** Show that if  $t \models \bigwedge_{yRx} \neg y$  then  $t \models x$ . This again follows from (C1).

**Axiom (b1).** Show that if  $t \models \neg x$  then  $t \models \bigvee_{yRx} y$ .

This follows from (C2). If  $t \models \bigvee_{yRx} y$  then  $t \models y_0$  for some  $y_0$ . So  $h_\lambda(y_0) = (\top, \top)$ . So  $\lambda(y_0) = \text{in}$ . So  $\lambda(x) = \text{out}$ . So  $h_\lambda(x) = (\perp, \perp)$  so  $t \models \neg x$

**Axiom (b2).** Show that if  $t \models \bigvee_{yRx} y$  then  $t \models \neg x$ . This again follows from (C2). ■

### 3 Refinements and remarks

{541-R3}

**Remark 3.1** We used the constant  $\mathbf{n}$ . We now try to eliminate it. Let  $\mathcal{A} = (S, R)$  be given. Define  $\mathbf{n}$  to be

$$\mathbf{n} = \bigwedge_{x \in S} (x \vee \neg x).$$

Assume for the moment that we are dealing with extensions which are not stable. So there exist undecided elements in the extension  $\lambda$ , i.e. there exists some  $x$  such that  $\lambda(x) = \text{und}$  and so  $h(x) = (\perp, \top)$ . So  $x \vee \neg x$  is false at  $t$  and true in  $s$ . So  $\mathbf{n}$  is false at  $t$  and true in  $s$ . So we found the  $\mathbf{n}$  we need for our translation.

We now offer a translation  $\Theta_{\mathcal{A}}^\lambda$  of an intuitionistic theory corresponding to any labelling  $\lambda$ . The translation is done by case analysis on  $\lambda$ .

1. Case  $\lambda$  does not give value und., i.e.  $\lambda(x)$  is either in or out. This means that  $\lambda$  represents a stable extension. Let  $\Theta_{\mathcal{A}}^0 = \bigwedge_x (x \leftrightarrow \bigwedge_{yRx} \neg y)$ .
2. Case  $\lambda$  gives value undecided. Let  $\Theta_{\mathcal{A}}^1 = \Delta_{\mathcal{A}}$  where the  $\mathbf{n}$  which is used in  $\Delta_{\mathcal{A}}$  is  $\mathbf{n} = \bigwedge_{x \in S} (x \vee \neg x)$ .

Define  $h_\lambda$  as in Section 2. We have that  $\lambda$  is a complete extension iff  $\Theta_{\mathcal{A}}$  holds under  $h_\lambda$ .

The translation and the two case analyses described above also work in the other direction. Let  $h$  be a model. Define  $\lambda_h$  as in Section 2. We distinguish two cases.

**Case  $t \models_h \mathbf{n}$ .** In this case we look at  $\Theta_{\mathcal{A}}^0$  and indeed,  $t \models \Theta_{\mathcal{A}}^0$  iff  $\lambda_h$  is a stable complete extension.

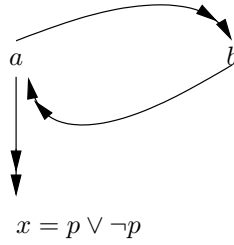


Figure 1:

{541-F5}

**Case  $t \not\models_h \mathbf{n}$ .** In this case look at  $\Theta_{\mathcal{A}}^1 = \Delta_{\mathcal{A}}$ . Indeed  $\lambda_h$  is a non-stable complete extension iff  $t \models_h \Delta_{\mathcal{A}}$ .

{541-R4}

**Remark 3.2** The perceptive reader might argue as follows:

*“What is the big deal about interpreting abstract argumentation in intuitionistic logic  $\mathbf{G}_3$ ? There are many such translations into many diverse systems. Your own paper [1] gave another interpretation in classical logic. Grossi has one in modal logic, there are many others as indeed quoted in [1]. What is really going on here is that the Caminada labelling requires 3 values, in, out and undecided. Any logic which can isolate 3 values for “in”, “out” and “undecided” can do the job! So what is the big deal here?”*

*The answer to the question of what is the “big deal” or added value or difference between translations lies in allowing for different possibilities for generalisations, extensions, variations, instantiations, or any other operations we do on argumentation networks. Different translations would react differently in different environments and interpretations.*

*Let us illustrate by an example. Take the two very similar translations proposed in our own papers, one is the translation into classical logic of [1] and the other is the present translation into intuitionistic logic .*

*We will show a substantially different behaviour in the context of instantiation. Instantiation is important, as any supporter of ASPIC ([15, 16, 17, 18]) will tell you. Let us take a very simple network with  $S = \{a, b, x\}$  and  $R = \{(a, b), (b, a), (a, x)\}$ . Let us instantiate  $x$  with  $x = p \vee \neg p$ . Consider Figure 1*

*In this figure,  $x$  is instantiated by the wff  $p \vee \neg p$ . When we translate into classical logic,  $p \vee \neg p$  is  $\top$  and so there can be only one extension.*

$$x = \top = \text{in}, b = \text{out}, a = \text{in}.$$

*In intuitionistic logic,  $\mathbf{G}_3, p \vee \neg p$  can be  $(\top, \top) = \text{in}$  or  $(\perp, \top) = \text{und}$ . So the possible extensions are*

$$\begin{aligned} a = \text{in}, b = \text{out}, x = \top = \text{in}, \\ b = \text{in}, a = \text{out}, x = \text{out}, \\ a = b = x = \text{und}. \end{aligned}$$

For further benefits see the next Remark 3.3 and Sections 4 and 5 below.

{541-R6}

**Remark 3.3** We assumed that  $(S, R)$  is finite. this is not necessary. We can assume that  $(S, R)$  is finitary, i.e. each  $x$  has a finite number of attackers. This will allow us to write  $\Delta_{\mathcal{A}}$  as an infinite theory containing wffs for each  $x \in S$ .

The correspondence between extensions  $\lambda$  and models  $h$  will still work.

The challenge is when we have nodes with an infinite number of attackers. We cannot write the expression like this:  $\bigwedge_{yRx} \neg y$ . It is not finite. For this we need a predicate theory talking about domains  $S$  with a binary relation  $xRy$  on  $S$  and a unary predicate  $In(x)$  on  $S$  and write

$$\forall y(yRx \rightarrow \neg In(y)).$$

The beauty of this language is that we can now also write

$$In(z) \rightarrow \neg xRy$$

which means that  $z$  attacks the attack from  $x$  to  $y$ . This is a higher level attack. Furthermore, since  $xRy$  is a formula of the logic, it can get value  $(\top, \top)$ , i.e. it is in for both worlds; or  $(\perp, \top)$ , i.e. it is undecided (out in the first but in in the second); or  $(\perp, \perp)$ , i.e. it is out for both worlds. The next section will do this systematically.

Furthermore, since  $xRy$  is a formula of the language, it can also go on and attack, so we can write an attack attacking another attack:

$$xRy \rightarrow \neg uRv$$

and an attack attacking another node:

$$xRy \rightarrow \neg In(z)$$

The reader might ask, if we have the relation  $R$  in the language, should we not have the following equivalence?

$$xRy \leftrightarrow (In(x) \rightarrow \neg In(y))$$

The answer is negative.  $xRy$  represents a geometrical relation in the network. It has a role in defining the attack conditions and the defining clauses for {in, out, und} values. The formula  $In(x) \rightarrow \neg In(y)$  also participates in these defining clauses, but describes the properties of the geometrical relation as a whole. So for example we do want that

$$xRy \rightarrow (In(x) \rightarrow \neg In(y))$$

but the converse does not necessarily hold. For example suppose  $(S, R)$  is as in Figure 2. Then  $In(a) \rightarrow \neg In(d)$  but it is not the case that  $aRd$ .

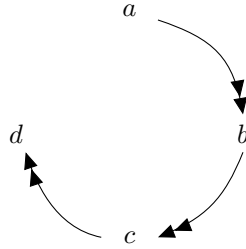


Figure 2: If  $a$  is in then  $b$  is out so  $c$  is in and  $d$  is out, but  $a$  does not attack  $d$ . {541-mike1}

## 4 The attack in Predicate Intuitionistic Logic.

{sec4}

This section informally develops the ideas put forward in Remark 3.3. The formal development will be in the next Section 5. We adopt the language of intuitionistic predicate logic with the usual connectives and quantifiers  $\{\wedge, \vee, \rightarrow, \neg, \perp, \top, \forall, \exists\}$  and the axioms for constant domains, namely:

$$\forall x(A \vee B(x)) \rightarrow A \vee \forall xB(x)$$

where  $x$  is not free in  $A$ .

Our logic, which we call  $\pi\mathbf{G}_3$ , is defined semantically using a two world predicate Kripke model of constant domains. The worlds are as before  $\{t, s\}$ , with  $t < s$  and  $t$  the actual world. We denote the domain of these worlds by  $S$ . This will also be our set of arguments. We allow two non-logical predicates in this language, a binary predicate  $xRy$  and a unary predicate  $In(z)$ .  $R$  represents the attack relation and  $In$  represents being “in” the extension defined by the model.

Being an intuitionistic language, any formula  $A$  has three options for values in the two world model:

- $t \not\vdash A$  and  $s \not\vdash A$
- $t \not\vdash A$  and  $s \vdash A$
- $t \vdash A$  and  $s \vdash A$

We can write the values as  $A = (\perp, \perp)$ ,  $A = (\perp, \top)$  and  $A = (\top, \top)$ .

For the intuitionistic logic and semantics see the Appendix.

{541-4R1}

**Remark 4.1** *This remark explains how the language with the predicates  $In(z)$  and  $xRy$  can deal with infinite networks.*

*Let  $(S, R)$  be an arbitrary network. So nodes  $x \in S$  may have an infinite number of attackers. We want to use predicate logic to define the analogous formulas to those of Definition 2.1.*

*We let  $\mathbf{n}$  be a constant as before whose truth value is  $(\perp, \top)$ . We now define the analogous wffs (A1), (A2), (B1), (B2). We take a model with domain  $S$ .*



We assume of this model that it has a constant domain and we assume that  $R$  is decided, namely:

$$\forall xy(xRy \vee \neg xRy)$$

The attack relation  $R$  is expressible in the model because  $R$  is in the language. We write:

$$(A1) \quad \forall x [In(x) \rightarrow (\mathbf{n} \vee \forall y (yRx \rightarrow \neg In(y)))]$$

$$(A2) \quad \forall x [\forall y (yRx \rightarrow \neg In(y)) \rightarrow (\mathbf{n} \vee In(x))]$$

$$(B1) \quad \forall x [\neg In(x) \rightarrow (\mathbf{n} \vee \exists y (yRx \wedge In(y)))]$$

$$(B2) \quad \forall x [\exists y (yRx \wedge In(y)) \rightarrow (\mathbf{n} \vee \neg In(x))]$$

Then we get the following theorem 4.2.

{541-4T2}

**Theorem 4.2** 1. Let  $\mathbf{m}$  be a model of  $\Delta_{\mathcal{A}}$ . Define  $\lambda_{\mathbf{m}}$  as follows:

- $\lambda_{\mathbf{m}}(x) = in$ , if  $\mathbf{m}(In(x)) = (\top, \top)$
- $\lambda_{\mathbf{m}}(x) = out$ , if  $\mathbf{m}(In(x)) = (\perp, \perp)$
- $\lambda_{\mathbf{m}}(x) = und$ , if  $\mathbf{m}(In(x)) = (\perp, \top)$ .

Then  $\lambda_{\mathbf{m}}$  is a complete extension of  $(S, R)$ .

2. Let  $\lambda$  be a complete extension of  $(S, R)$ . Define a model  $\mathbf{m}$  with domain  $S$  and a decided relation  $R$ . Define the values for  $In(x)$ ,  $x \in S$  as follows:

- $In(x) = (\top, \top)$ , if  $\lambda(x) = in$
- $In(x) = (\perp, \perp)$ , if  $\lambda(x) = out$
- $In(x) = (\perp, \top)$ , if  $\lambda(x) = und$

Then  $\mathbf{m}$  is a model of  $\Delta_{\mathcal{A}}$ .

**Proof.** Parallel to the proof of Theorem 2.2. ■

{541-4R3}

**Remark 4.3** The inclusion of  $R$  in the predicate language gives us the opportunity to express higher level attacks not only on nodes and other attack arrows but also on meta-level statements. Consider the higher level network of Figure 3. In this figure we have:

- (i)  $a \rightarrow b$
- (ii)  $a \rightarrow c$
- (iii)  $a \rightarrow (c \rightarrow d)$
- (iv)  $(a \rightarrow b) \rightarrow d$
- (v)  $c \rightarrow d$

(i) and (ii) and (v) are ordinary attacks. (iii) is a higher level attack first

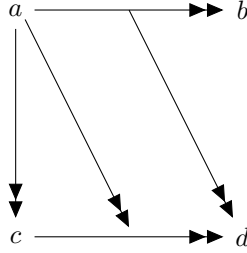


Figure 3:

{541-4F4}

introduced in [14] and further studied extensively in [10], [11], [12] and [13]. (iv) was introduced in [14] and has never had systematic semantics until the current paper.

The above attacks can be translated using  $In(z)$  and  $xRy$  as follows: we need to give up the axiom  $\forall xy(xRy \vee \neg xRy)$  and allow  $xRy$  to get the undecided value  $(\perp, \top)$ . Because of the possibility that an attack arrow  $xRy$  may be out or undecided we need to express explicitly in the clauses below that the relation holds, we can now write:

- (i')  $(In(a) \wedge aRb) \rightarrow \neg In(b)$
- (ii')  $(In(a) \wedge aRc) \rightarrow \neg In(c)$
- (iii')  $In(a) \rightarrow \neg cRd$
- (iv')  $aRb \rightarrow \neg In(d)$
- (v')  $(In(c) \wedge cRd) \rightarrow \neg In(d)$

We note the important aspect of this intuitionistic modelling is that any wff can be subject to attack. For example take the meta-level statement

$$\exists y(dRy)$$

this is false in the network. In the syntax it can be attacked by any  $z$ :

$$In(z) \rightarrow \neg \exists y(dRy)$$

We can extend clauses (A1), (A2), (B1), (B2) to encompass any attacks on a wff  $W$  by replacing “ $In(x)$ ” by “ $W$ ”, and  $\{“In(y)” \mid yRx\}$  by the wffs  $\{Y \mid Y \rightarrow W\}$  (i.e. the set of  $Y$  that attack  $W$  according to our network).

{541-4D5}

**Example 4.4** Let us continue and analyse the higher level network of Figure 3.

The network can be described as  $(S, \mathbb{R})$ , where  $S = \{a, b, c, d\}$  and the relation  $\mathbb{R} \subseteq (S \cup S^2)^2$  is:

$$\{(a, b), (a, c), (c, d), (a, (c, d)), ((a, b), d)\}$$

corresponding to

$$\{a \rightarrow b, a \rightarrow c, c \rightarrow d, a \rightarrow (c \rightarrow d), (a \rightarrow b) \rightarrow d\}$$

ht

Attack	representation
$a \rightarrow b$	$(In(a) \wedge aRb) \rightarrow \neg In(b)$
$a \rightarrow c$	$(In(a) \wedge aRc) \rightarrow \neg In(c)$
$c \rightarrow d$	$(In(c) \wedge cRd) \rightarrow \neg In(d)$
$a \rightarrow (c \rightarrow d)$	$In(a) \rightarrow \neg cRd$
$(a \rightarrow b) \rightarrow d$	$aRb \rightarrow \neg In(d)$

Table 1:

{541-4F6}

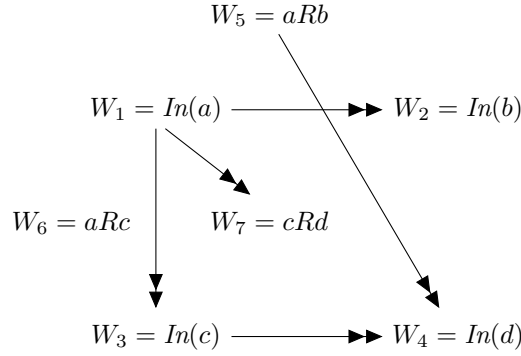


Figure 4:

{541-4F7}

Because we have an attack predicate  $R$  in the intuitionistic language we can express  $\mathbb{R}$  using  $R$ .

To do this effectively we need to define the set  $\mathbb{W}$  of elements (units) which can participate in the higher level attacks. These are formulas of intuitionistic logic built up from Figure 3. Let us list them:

- The set  $\mathbb{W}$ :
- ( $W_1$ )  $In(a)$
  - ( $W_2$ )  $In(b)$
  - ( $W_3$ )  $In(c)$
  - ( $W_4$ )  $In(d)$
  - ( $W_5$ )  $aRb$
  - ( $W_6$ )  $aRc$
  - ( $W_7$ )  $cRd$

The attack relation  $\mathbb{R}$  can now be expressed in the object level of the language with  $In$  and  $R$ . The set  $\mathbb{W}$  with elements ( $W_1$ )-( $W_7$ ) together with Table 1 gives rise to the argumentation network of Figure 4. We need axioms to make this work. Axioms (a1), (a2), (b1) and (b2) now use  $\mathbb{W}$  and Table 1. They become the following (fa1), (fa2), (fb1), (fb2) and (fc). The construction uses Figure 4 and the attack relation of Table 1, which we can call  $(\mathbb{W}, \mathbb{R}_1)$ . We are now in the situation of Section 2 Definition 2.1.

The meaning of the clauses of Definition 2.1 in terms of attacks and objects participating in attacks are as follows.

$$\begin{aligned}
\text{meaning of } a1: & \bigwedge_{\text{all objects } W} (W \text{ is in} \rightarrow (\mathbf{n} \vee \bigwedge_{\text{attackers } Y \text{ of } W} Y \text{ is out})) \\
\text{meaning of } a2: & \bigwedge_{\text{all objects } W} (\bigwedge_{\text{attackers } Y \text{ of } W} Y \text{ is out} \rightarrow (\mathbf{n} \vee W \text{ is in})) \\
\text{meaning of } b1: & \bigwedge_{\text{all objects } W} (W \text{ is out} \rightarrow (\mathbf{n} \vee \bigvee_{\text{attackers } Y \text{ of } W} Y \text{ is in})) \\
\text{meaning of } b2: & \bigwedge_{\text{all objects } W} (\bigvee_{\text{attackers } Y \text{ of } W} Y \text{ is in} \rightarrow (\mathbf{n} \vee W \text{ is out}))
\end{aligned}$$

Table 2:

{541-MT1}

If we follow the above meaning and use  $(\mathbb{W}, \mathbb{R}_1)$ , we get the following:

$$\begin{aligned}
(\text{fa1}) \quad & (In(b) \rightarrow (\mathbf{n} \vee \neg In(a) \vee \neg aRb)) \\
& \wedge (In(c) \rightarrow (\mathbf{n} \vee \neg In(a) \vee \neg aRc)) \\
& \wedge (In(d) \rightarrow (\mathbf{n} \vee \neg In(c) \vee \neg cRd)) \\
& \wedge (cRd \rightarrow (\mathbf{n} \vee \neg In(a))) \\
& \wedge (In(d) \rightarrow (\mathbf{n} \vee \neg aRb))
\end{aligned}$$

$$\begin{aligned}
(\text{fa2}) \quad & ((\neg In(a) \vee \neg aRb) \rightarrow (\mathbf{n} \vee In(b))) \\
& \wedge ((\neg In(a) \vee \neg aRc) \rightarrow (\mathbf{n} \vee In(c))) \\
& \wedge ((\neg In(c) \vee \neg cRd) \rightarrow (\mathbf{n} \vee In(d))) \\
& \wedge (\neg In(a) \rightarrow (\mathbf{n} \vee cRd)) \\
& \wedge (\neg aRb \rightarrow (\mathbf{n} \vee In(d)))
\end{aligned}$$

$$\begin{aligned}
(\text{fb1}) \quad & (\neg In(b) \rightarrow (\mathbf{n} \vee (In(a) \wedge aRb))) \\
& \wedge (\neg In(c) \rightarrow (\mathbf{n} \vee (In(a) \wedge aRc))) \\
& \wedge (\neg In(d) \rightarrow (\mathbf{n} \vee (In(c) \wedge cRd))) \\
& \wedge (\neg cRd \rightarrow (\mathbf{n} \vee In(a))) \\
& \wedge (\neg In(d) \rightarrow (\mathbf{n} \vee aRb))
\end{aligned}$$

$$\begin{aligned}
(\text{fb2}) \quad & ((In(a) \wedge aRb) \rightarrow (\mathbf{n} \vee \neg In(b))) \\
& \wedge ((In(a) \wedge aRc) \rightarrow (\mathbf{n} \vee \neg In(c))) \\
& \wedge ((In(a) \wedge cRd) \rightarrow (\mathbf{n} \vee \neg In(d))) \\
& \wedge (In(a) \rightarrow (\mathbf{n} \vee \neg cRd)) \\
& \wedge (aRb \rightarrow (\mathbf{n} \vee \neg In(d)))
\end{aligned}$$

The reader will notice that we actually used conjunctive attacks in this example. Figure 4 really should be Figure 5. We shall discuss this formally in the next section 5.

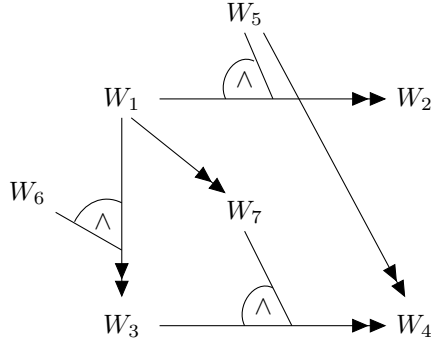


Figure 5: Our notation for joint attacks is as in Figure 6

{541-4F7a}

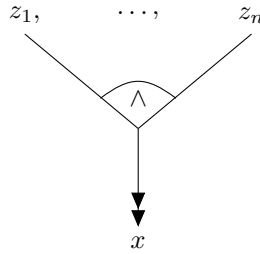


Figure 6: The reading is  $\bigwedge_i In(z_i) \rightarrow \neg In(x)$

{541-4F8}

## 5 Formal development of the intuitionistic model

{sec5}

We begin with further motivation for our meta-level view of argumentation networks. Let us start with a traditional network  $(S, R)$  with  $R \subseteq S \times S$ , and let us focus on the following two types of activities associated with it.

### 1. Instantiation

We instantiate elements  $x$  of  $S$  by arguments from some logical system  $\mathbf{L}$ . This case is the ASPIC type of instantiation (see [15, 16, 17, 18]). In all of these instantiations,  $\mathbf{L}$  is *not* based on the language of  $(S, R)$  itself. To be quite clear, we deal with wffs which talk about some application area where the arguments of  $S$  come from, and are not about the geometry of  $(S, R)$  itself.

There are arguments, however, which use the geometry of  $(S, R)$ . Such arguments are, for example, the formula  $W(a)$ , mentioned in section 1 and other arguments like

$J(a, b)$  : The arguments  $a$  and  $b$  attack the same arguments.

Obviously they are not independent (think of two witnesses co-ordinating their testimony).

$$J(a, b) = \forall x(aRx \leftrightarrow bRx)$$

## 2. Abduction

Abduction in argumentation is a meta-level operation on  $(S, R)$  which can take two forms:

- (a) *Observation*: for example, we can ask if there are extensions in which  $x$  is in or  $x$  is out, etc.
- (b) *Intervention*: for example, the instruction to delete  $x$  from  $S$  or to ensure  $x$  is out (or in), or the instruction to delete (or ignore) all self-attacking arguments, or the like.

Observation is easy to implement in our intuitionistic framework. Given  $\mathcal{A} = (S, R)$ , if we want to observe whether  $\phi$  holds, we look for models of  $\Delta_{\mathcal{A}} \cup \{\phi\}$ . So if  $x \in S$ , we can take  $\phi$  to be  $In(x)$  or  $\neg In(x)$ , and so on. Since every model of  $\Delta_{\mathcal{A}}$  is an extension, then the models of  $\Delta_{\mathcal{A}} \cup \{\phi\}$  would yield extensions satisfying  $\phi$ , if they exist.

Intervention is more of a problem. How do we intervene and delete, or ignore, an argument  $a \in S$ ? We can add to  $S$  a new annihilator node  $x_a$  with  $x_a R a$ . This will ensure that  $a$  is out in  $(S \cup \{x_a\}, R \cup \{(x_a, a)\})$ .

We may, however, want to intervene in a more subtle way. Say we want to delete every element which attacks every other element. We can do that by identifying such elements  $a$  and adding an annihilator  $x_a$  for each such  $a$ . This is not subtle at all. It does not give  $a$  a chance to protect or defend itself. To make this case more concrete, suppose  $a$  is an argument which attacks everything opposed to some ‘‘Holy Book’’. We might want to intervene and take  $a$  out because we do not want such considerations to be involved. However, there may be arguments against such intervention. We may not want to conduct the discussion of whether to intervene or not in the meta-level. Furthermore, the discussion may involve object level considerations as well as meta-level considerations. For example,  $a$  itself may contain counter argument intended to forestall intervention, or there may be an argument that if we were to take  $a$  out we might get the wrong extensions, etc.

So how do we allow  $(S, R)$  to include such arguments and let them attack? In the above example we must let  $x_a$  be something like the formula below, which actually states the reason why we want to intervene and take  $a$  out:

$$x_a = \forall y (y \neq a \rightarrow a R y)$$

and add  $x_a$  to  $S$  with  $x_a R a$ . In that case, we can allow  $a$  to protect itself by saying  $J(a)$ :

$$J(a) = \forall x (a R x \leftrightarrow x R x)$$

i.e.  $a$  defends itself by saying that it attacks self-attacking arguments (e.g. because the Holy Book is not tolerant of them).

The above discussion favours argumentation networks containing formulas of the predicate logic of  $R$ . We thus have as arguments the set  $(S \cup W, R)$  with  $R \subseteq (S \cup W)^2$  and  $W$  is a set of wffs in the logic  $\mathbf{L}(R)$  based on the binary relation  $R$ .

If we think more along these lines, we see that we have a challenge for ASPIC itself. We know it is perfectly reasonable to discuss what arguments to allow into  $(S, R)$ . So let us do the ASPIC approach and use this discussion as the generator of the arguments for  $(S, R)$ . We thus have a loop here, how do we give semantics to the process?

Our intuitionistic logic can do this easily. The wffs of  $\mathbf{L}(R)$  are available in the language. So:

- If  $\phi$  attacks  $\psi$  we write  $\phi \rightarrow \neg\psi$
- If  $\phi$  attacks  $x$  we write  $\phi \rightarrow \neg In(x)$
- If  $x$  attacks  $y$  we write  $(In(x) \wedge xRy) \rightarrow \neg In(y)$
- If  $x$  attacks  $\phi$  we write  $In(x) \rightarrow \neg\phi$ .

$\phi$  and  $\psi$  can also be  $xRy$  or  $uRv$  and thus we can also have higher level attacks.

**Example 5.1** *Let us give a natural example of a network where a formula  $\phi(R)$  attacks other arguments. Consider a network  $(S, R)$  where  $S = \{a, x, y, z\}$  and where each of  $x, y, z$  attacks  $a$  (Figure 7). In this Figure,  $a$  is attacked by all other arguments in the network (i.e. by everything else in  $S$ ). This state of affairs is described by  $\phi$  which can be used to protect  $a$  (extending  $R$ ). In this case  $\phi$  can be seen as discrediting attacks as being part of an unfair, universal, attack on  $a$ .*

{541-EAug17-0}

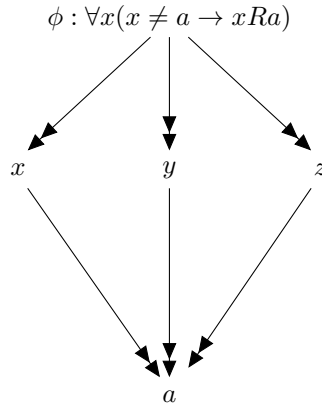


Figure 7:

{541-FAug17-1}

*The situation is similar in Figure 8, where  $S = \{a, b, x, y\}$  and where  $b$  attacks  $a, x,$  and  $y$ .  $\phi$  protects  $a$  from  $b$  by attacking it on the grounds that it attacks everything.*

*Such arguments are very common in natural reasoning. We call them “would-would’nt” arguments. For example, the head of the opposition party might strongly attack the policies of the current government. The head of the current*

government might defend his policies by saying “well the leader of the opposition would say all of that wouldn’t he, he always attacks everything I say, because he is the leader of the opposition”.<sup>3</sup>

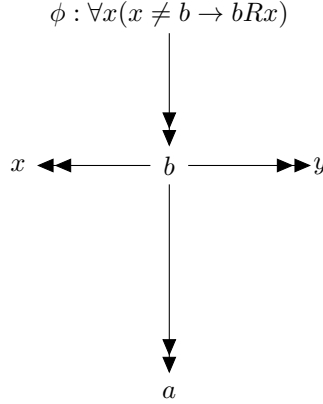


Figure 8:

{541-FAug17-2}

**Example 5.2** Consider Figure 9. This figure presents a problem. Since  $a$  is

{541-5E1}

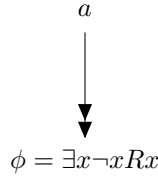


Figure 9:

{541-5F2}

not attacked, we have  $a = in$  i.e. the value of  $In(a)$  is  $(\top, \top)$ . Thus  $\phi$  must be out, i.e.  $\phi = (\perp, \perp)$ . But if  $\exists x \neg xRx$  is false, we get that we must have  $aRa$  and so how can  $a$  be in? Then we must go for  $In(a) = und = (\perp, \top)$ . So  $\phi$  is also und,  $\phi = (\perp, \top)$ . So at the world  $s$  we have  $In(a) \rightarrow \mathbf{n} \vee \exists \neg xRx$ . This holds.

Let us look at Figure 9 slightly differently. The domain of the Figure is  $S = \{a\}$ . So “ $\exists x \phi(x)$ ” simply means  $\phi(a)$ . So  $\exists x \neg xRx$  is just  $\neg aRa$ . Now a attacking  $\neg aRa$  is  $In(a) \rightarrow \neg \neg aRa$ . This is also possible because the translation into intuitionistic logic says  $In(a) \rightarrow \mathbf{n} \vee \neg \neg aRa$ .

It looks like we need to accept that although a node  $x$  may not be attacked geometrically, it may be attacked by some  $\phi$  in the network by virtue of its meaning. The simplest example is Figure 10. In Figure 10 the node  $aRa$  does not attack the node  $a$  geometrically, but they cannot both be in.

The situation, however, is still paradoxical. In Figure 9, “ $\phi$ ”= $\exists x \neg xRx$  and “ $a$ ” are consistent together. The problem arises because  $a$  attacks  $\phi$ . We can

<sup>3</sup>Thanks to L. Rivlin for pointing out this case.



$a \qquad aRa$

Figure 10:

{541-5F3}

consider the database  $\{a, \exists x \neg xRx, a \rightarrow \neg \exists x \neg xRx\}$ . This database describes the figure: for example if  $a$  is in then it does not attack itself, and so  $\exists x \neg xRx$  is also in. It is inconsistent. So perhaps we resolve the inconsistency by having the attack “out”, i.e. we cancel the attack  $a \rightarrow \neg \phi$ .

The above example impresses upon us the need simply to “run” the translation into intuitionistic logic and see what it does. If we can accept what it does as reasonable we can proceed from there, otherwise we analyse why we find the result unacceptable and seek an improvement. But our next step must be now to check the translation first.

{541-5R5}

**Remark 5.3** *Let us analyse more closely the problems with the network of Figure 10.*

1. *The network has an unattacked element  $aRa$  in it. Therefore  $aRa$  is in. But being in, it changes the original network. Now  $a$  attacks itself, so how can  $a$  also be in?*
2. *Suppose we adopt the remedy that we include only wffs  $\phi$  that are true in  $(S, R)$ . If this is the case then  $\phi = \top$ , so how can  $\phi$  ever be out, or how can  $\phi$  ever be counter-attacked? So this ‘remedy’ is not a good solution.*
3. *So what is the source of the problem? The problem arises because when we construct the theory  $\Delta_{\mathcal{A}}$ , we look at  $\mathcal{A} = (S \cup W, R)$ , with  $R \subseteq (S \cup W)^2$ , and get the complete extensions as model of  $\Delta_{\mathcal{A}}$ . In such models however,  $R$  changes to  $R'$  and maybe even  $S$  changes to  $S'$  (if some  $\exists \phi(x)$  is in  $W$  and is in). So we lose the correct representation theorem. So we must calculate the complete extension relative to  $(S' \cup W, R')$ . This means that  $\Delta_{\mathcal{A}}$  must be changed. Can we do that implicitly in one move? We need to write  $\Delta_{\mathcal{A}'}$  for  $\mathcal{A}' = (S', R')$ , when we do not know what  $\mathcal{A}'$  is!*
4. *Let us see how it is done for our example of Figure 10, perhaps we can get some new ideas. Since  $aRa$  is in, the network becomes  $\mathcal{A}'$ , Figure 11. So we need to write  $\Delta_{\mathcal{A}'}$  and not  $\Delta_{\mathcal{A}}$ . However, when we wrote the intuitionistic theory for Figure 10, we wrote  $\Delta_{\mathcal{A}}$ , with, for example clause (a2), by looking at Figure 10 and not by looking at Figure 11. We wrote:*

$$\bigwedge_{\text{attackers } y \text{ of } a} y \text{ is out} \rightarrow (\mathbf{n} \vee a \text{ is in})$$

Since in Figure 10 there are no attackers of  $a$  and the empty conjunction is  $\top$ , we got  $\mathbf{n} \vee In(a)$ . Had we looked at Figure 11 we would have had

$$(\neg In(a) \vee \neg aRa) \rightarrow (\mathbf{n} \vee In(a))$$



Figure 11:

{541-5F6}

and the solution is  $In(a) = (\perp, \top)$ .

The problem is that we do not know in general what  $R'$  is going to be until we compute the extension, so we do not know what to write in order to compute the extension. So we ask, are we in a catch-22 situation? The answer is that there is a solution. Since we have the predicate  $xRy$  in the language, we do not need to look at the new Figure 11. We simply adopt the view that every node  $x \in S$  attacks every node  $y \in S$  by the joint attack  $In(x) \wedge xRy$ , independent of whether it is shown in  $R$  or not. So for each node  $x \in S$ , to say that it is not attacked by other nodes, we write:

$$\bigwedge_{y \in S} (\neg In(y) \vee \neg yRx).$$

To say, in addition, that  $x$  is not attacked by any  $\phi$  such that  $\phi Rx$ , we also add  $\bigwedge_{\phi Rx} \neg \phi$  to the main conjunct. So (a2) becomes (a2\*):

$$(a2^*) \quad \left( \left( \left( \bigwedge_{\phi Rx} \neg \phi \right) \wedge \left( \bigwedge_{yRx} (\neg In(y) \vee \neg yRx) \right) \right) \rightarrow (\mathbf{n} \vee In(x)) \right) \wedge \bigwedge_{\psi \in W} \left( \left( \bigwedge_{\phi R\psi} \neg \phi \right) \wedge \left( \bigwedge_{yR\psi} (\neg In(y) \rightarrow (\mathbf{n} \vee \psi)) \right) \right)$$

Any model of the theory  $\Delta_{A'}$  will tell us what  $R$  is and that will make (a2\*) behave correctly.

Let us do this for Figure 9.

(a1) becomes (a1\*):

$$(In(a) \rightarrow (\mathbf{n} \vee \neg In(a) \vee \neg aRa)) \wedge (\exists x \neg xRx \rightarrow (\mathbf{n} \vee \neg In(a)))$$

(a2) becomes (a2\*):

$$((\neg In(a) \vee \neg aRa) \rightarrow (\mathbf{n} \vee In(a))) \wedge (\neg In(a) \rightarrow (\mathbf{n} \vee \exists x \neg xRx))$$

(b1) becomes (b1\*):

$$(\neg In(a) \rightarrow (\mathbf{n} \vee \neg In(a) \vee aRa)) \wedge (\neg \exists x \neg xRx \rightarrow (\mathbf{n} \vee In(a)))$$

(b2) becomes (b2\*):

$$((In(a) \wedge aRa) \rightarrow (\mathbf{n} \vee \neg In(a))) \wedge (In(a) \rightarrow (\mathbf{n} \vee \neg \exists x \neg xRx))$$

Let us now check what models this theory can have. Let us write the axioms from the point of view of the world  $t$  (the actual world).  $\mathbf{n}$  is false at  $t$  so we get:

**Axioms (a1\*) at  $t$ :**

- $t \models In(a) \Rightarrow t \models \neg In(a) \vee \neg aRa$
- $t \models \exists x \neg xRx \Rightarrow t \models \neg In(a)$

**Axioms (a2\*) at  $t$ :**

- $t \models \neg In(a) \vee \neg aRa \Rightarrow t \models In(a)$
- $t \models \neg In(a) \Rightarrow t \models \exists x \neg xRx$

**Axioms (b1\*) at  $t$ :**

- $t \models \neg In(a) \Rightarrow t \models In(a) \wedge aRa$
- $t \models \neg \exists x \neg xRx \Rightarrow t \models In(a)$

**Axioms (b2\*) at  $t$ :**

- $t \models In(a) \wedge aRa \Rightarrow t \models \neg In(a)$
- $t \models In(a) \Rightarrow t \models \neg \exists x \neg xRx$

We now check what models this theory  $\Delta = \{(a1^*), (a2^*), (b1^*), (b2^*)\}$  can have. Suppose we have a model, what values can  $In(a)$  take?

**Case 1.**  $In(a) = (\top, \top)$ . From (a1\*) we get  $t \models \neg aRa$ , but this contradicts (b2).

**Case 2.**  $In(a) = (\perp, \top)$ . From (b1\*) we get that  $t \not\models \neg \exists x \neg xRx$  and so  $s \models \exists x \neg xRx$ . Can  $t \models \exists x \neg xRx$ ? From (a1\*) we get that if  $t \models \exists x \neg xRx$  we would have  $t \models \neg In(a)$ , which contradicts  $In(a) = (\perp, \top)$ . So  $t \not\models \exists x \neg xRx$  and so  $\exists x \neg xRx = (\perp, \top)$ .

Do we have a model? Do all axioms hold for  $In(a) = \exists x \neg xRx = (\perp, \top)$ ? We check and the answer is “yes”.

**Case 3.**  $In(a) = (\perp, \perp)$ . This contradicts  $(a2^*)$ .

{541-5F7}

**Remark 5.4 (Intermediate evaluation)** Let us see what we have so far. We have a procedure as follows:

1. Given a network of the form  $\mathcal{A} = (S \cup W, R')$  with  $S$  a set of nodes and  $W$  a set of wff in the language  $\mathbf{L}(R)$  of a binary relation  $R$  and  $R' \subseteq (S \cup W)^2$ , we construct a theory  $\Delta_{\mathcal{A}'}$  on the network  $\mathcal{A}'$  using clauses  $(a1^*)$ ,  $(a2^*)$ ,  $(b1^*)$ , and  $(b2^*)$  where these clauses are constructed according to the meanings given by Table 2 of Example 4.4.
2. We can define the complete extensions for  $\mathcal{A}'$  as the models of  $\Delta_{\mathcal{A}'}$ . This definition is motivated by our discussions so far. We need to make some soundness checks and if we can, add a representation theorem reducing this proposed definition to some familiar terms.

Let us do a quick soundness/plausibility check for this definition. Suppose we have no  $W$ , i.e.  $W = \emptyset$ . Then our network is  $\mathcal{A} = (S, R)$  with  $R \subseteq S \times S$ .<sup>4</sup> We know how to find the complete extensions for  $\mathcal{A}$ , as explained in Section 2 in Definition 2.1 and we have the Representation Theorem 2.2 showing the correctness of the translation, into propositional  $\mathbf{G}_3$ , of clauses  $(a1)$ ,  $(a2)$ ,  $(b1)$  and  $(b2)$ . These clauses are propositional and do not use any predicates. But now we can also look at  $(S, R)$  as an argument network with  $W$  empty and so we need to look at  $\mathcal{A}$  as  $\mathcal{A}' = (S, R)$ . This is to be considered as another network which is translated into predicate intuitionistic logic  $\pi\mathbf{G}_3$ , with the predicates  $In$  and  $R$  in the language. This translation uses the clauses  $(a1^*)$ ,  $(a2^*)$ ,  $(b1^*)$  and  $(b2^*)$  which follow the meanings given by Table 2.

Do we get the same extensions? The answer is “yes”. We will not give a proof but an explanation. Consider for example clause  $(a2)$ . It says:

$$(a2) \quad \bigwedge_x \left( \bigvee_{yRx} y \rightarrow (\mathbf{n} \vee \neg x) \right)$$

In predicate logic this clause becomes:

$$(A2) \quad \forall x (\exists y (yRx \wedge In(y)) \rightarrow (\mathbf{n} \vee \neg In(x)))$$

we can agree that the two clauses say the same thing and this observation was indeed exploited in Section 4 Remark 4.1 and Theorem 4.2. The clause  $(a2^*)$  for our case of  $\mathcal{A}'$  becomes:

$$(a2^*) \quad \forall x \left( \bigvee_{y \text{ attacks } x} y \text{ is in} \rightarrow (\mathbf{n} \vee x \text{ is out}) \right).$$

Now since  $W = \emptyset$ , the attacks of  $x$  are just  $\{y \mid yRx\}$  and therefore  $(a2^*)$  and  $(A2)$  are the same.

So the definition has passed this quick soundness check.

<sup>4</sup>We use the letter  $\mathcal{A}$  (and not  $\mathcal{A}'$ ) when referring to  $(S, R)$  in order to indicate that we want to consider  $(S, R)$  as a traditional network. When we wish to indicate  $(S, R)$  is a network with  $W = \emptyset$  then we refer to it using  $\mathcal{A}'$ .

**Remark 5.5**

1. Let  $\mathcal{A} = (S_0, R_0)$  be a network with  $m$  elements. We can assume for simplicity that  $S_0 = \{1, 2, \dots, n\}$  and  $R \subseteq S_0^2$ . Let  $\mathbf{L}(R, In, =)$  be an intuitionistic language with unary  $In$  and binary  $R$  and the equality symbol  $=$ . We shall use  $\mathbf{L}(R, In, =)$  to describe  $(S_0, R_0)$  completely in  $\pi\mathbf{G}_3$  intuitionistic logic by a theory which we call  $\Omega_{\mathcal{A}}$ .

2.  $\Omega_{\mathcal{A}}$  has the following clauses:

(a) Axioms for equality  $=$ , together with the decidedness axioms:

$$\forall xy(x = y \vee \neg x = y)$$

$$\forall xy(xRy \vee \neg xRy)$$

(b)

$$\exists x_1, \dots, x_n \left( \forall y \left( \bigvee_{i=1}^n y = x_i \right) \wedge \Psi_{\mathcal{A}} \right)$$

where we set  $\Psi_{\mathcal{A}} = \bigwedge_{(i,j) \in R_0} x_i R x_j$ .

3. Let  $(A1)$ ,  $(A2)$ ,  $(B1)$  and  $(B2)$  be as in Remark 4.1 and let  $\Delta_{\mathcal{A}}$  be the theory  $\{(A1), (A2), (B1), (B2)\}$ .

4. The following holds:

- $h$  is a model of  $\Delta_{\mathcal{A}} \cup \Omega_{\mathcal{A}}$  iff  $\lambda_h$  is a complete extension of  $(S_0, R_0)$ .

**Remark 5.6** Continuing the discussion of Remark 5.5 we note that we can regard the network  $(S_0, R_0)$  of Remark 5.5 as a network  $\mathcal{A}' = (S_0 \cup \{\Omega_{\mathcal{A}}\}, R')$  in the sense of Remark 5.4, with  $R' = R_0$ . This network contains  $\Omega_{\mathcal{A}}$  unattacked. Therefore  $\Omega_{\mathcal{A}}$  holds in any model  $h$ , and the axioms of  $\Delta_{\mathcal{A}}$  make sure the model is a complete extension.

So  $\Omega_{\mathcal{A}}$  describes the network and  $\Delta_{\mathcal{A}}$  forces the models to be complete extensions of the network which  $\Omega_{\mathcal{A}}$  describes.

Viewed in this way, we notice that  $\Psi_{\mathcal{A}} = \bigwedge_{(i,j) \in R_0} x_i R x_j$  is embedded in the wff  $\Omega_{\mathcal{A}}$  and plays the role of describing the attack relation  $R_0$  of  $\mathcal{A}$ .  $\Psi_{\mathcal{A}}$  does this descriptive job extensionally, in a traditional manner, by listing what attacks what. In this set up, however, we can allow  $\Psi_{\mathcal{A}}$  to describe  $R$  axiomatically, not by listing the members of  $R$  but by writing axioms for  $R$ . For example we can take an axiom  $\phi$  to be:

$$\phi = \neg \exists x x R x \wedge \exists y \forall z \neg z R y.$$

$\phi$  says that  $R$  has no self attacking elements and at least one element that is not attacked. By letting  $\Psi_{\mathcal{A}} = \phi$  we are looking at a new type of generic networks of the form  $(S_0, \phi(R_0))$ , where  $R_0$  is restricted axiomatically by  $\phi$ .<sup>5</sup>

<sup>5</sup>Define the concept of an *axiomatic argumentation network* to be of the form  $(S_0, \phi(R_0))$  with  $\phi$  a wff describing properties of  $R$ . We consider this further in Section 6.

So to give an example, consider  $S_0$  as a set of two elements  $\{a, b\}$ .  $\phi$  allows for the following options for  $R_0$  (this is Example 6.3 of Section 6 below):

(i)  $R_0^1 = \emptyset$

(ii)  $R_0^2 = \{a \rightarrow b\}$

(iii)  $R_0^3 = \{b \rightarrow a\}$

The complete extensions are

(i\*)  $a=b=in$

(ii\*)  $a=in, b=out$

(iii\*)  $a=out, b=in$

Note that we can describe  $\phi$  equivalently in this case as

$$\phi = \bigvee_{i=1}^3 (R_0^i \wedge \bigwedge_{j \neq i} \neg R_0^j).$$

We further note that we can allow  $\phi$  to contain additional predicates besides  $R$  which help describe what kind of  $R$  we want. These additional predicates may come from an application context for  $R$  and may also include the predicate  $In$ . We shall say more on this in Section 6.

## 6 Axiomatic argumentation frames (AAF)

This section continues the ideas of Remark 5.5. We now motivate the new concept of *Axiomatic Argumentation Frames* (AAF) and give formal definitions.

{sec6}

**Example 6.1** Suppose we are given an argumentation network  $\mathcal{A} = (S, R)$  which is too large in the sense that  $S$  has too many members, and that we wish to select a subset of  $\mathcal{A}$ . For example, suppose we need to give a presentation and, for practical reasons (e.g. time, intelligibility etc.), we wish to select only a subset  $S'$  of  $S$  to appear in this presentation.<sup>6</sup> Let us suppose we have calculated that our presentation has time to discuss exactly  $k$  arguments. So we need to pick a suitable  $k$  sized subset  $S'$  from  $S$ .

{541-EAug1}

But not any  $S'$  will do, as we might have further conditions on what  $S'$  should look like: for example we might wish it to appear to present a coherent position say by having a nontrivial complete extension; or we might wish it to reflect  $S$  in some way; or we may have some other external criteria for selecting  $S'$  (such as degree of relevance to the topic). When choosing  $S'$  we will check whether these conditions are met by  $(S', R \upharpoonright S')$ . Ultimately, we will need to write a wff  $\Psi$  describing these conditions in predicate logic and then consider the sets  $(S_j, R \upharpoonright S_j)$  which satisfy  $\Psi$ .

<sup>6</sup>But we would wish the presentation to remain coherent view that is as self-contained as possible.

**Definition 6.2**

1. Let  $\mathbf{L}$  be a predicate language containing a binary predicate  $R$  and a unary predicate  $S$ . Let  $\Psi$  be a consistent classical theory of  $\mathbf{L}$ . Let  $\mathbf{M} = \mathbf{M}(\Psi)$  be a model of  $\Psi$ . Let  $S_{\mathbf{M}}$  be the extension of  $S$  in  $\mathbf{M}$  and let  $R_{\mathbf{M}}$  be the binary extension of the relation of  $\mathbf{M}$ . We ignore the extensions of all other predicates of  $\mathbf{L}$  which are involved in  $\mathbf{M}$ . We view them as parameters helping to write the axiomatic properties of  $S$  and  $R$  (via the theory  $\Psi$ ).
2. We can view  $\mathcal{A}_{\mathbf{M}} = (S_{\mathbf{M}}, R_{\mathbf{M}} \upharpoonright S_{\mathbf{M}})$  as a traditional argumentation network with complete extensions  $E_1^{\mathbf{M}}, \dots, E_{r(\mathbf{M})}^{\mathbf{M}}$ .
3. Let  $S_0$  be a finite set and let  $\{\mathbf{a} \mid a \in S_0\}$  be the set of names in  $\mathbf{L}$  for elements of  $S_0$ . Let  $\phi(S_0)$  be the formula

$$\phi(S_0) = \forall x \left( \bigvee_{a \in S_0} x = \mathbf{a} \right) \wedge \bigwedge_{\substack{a, b \in S_0 \\ a \neq b}} \mathbf{a} \neq \mathbf{b}.$$

Let  $\Psi(S_0) = \Psi \wedge \phi(S_0)$ .

4. We define the notion of the axiomatic argumentation network  $\mathcal{A}_0 = (S_0, \Psi(S_0))$  to be the family of networks obtained from models  $\mathbf{M}$  of  $\Psi(S_0)$  as defined in item (1) above. We also define the extensions of  $\mathcal{A}_0$  as all the extensions  $E_i^{\mathbf{M}}$  of all possible models  $\mathbf{M}$  of  $\Psi(S_0)$ . Note that because of the axiom  $\Psi(S_0)$ , all networks  $(S_{\mathbf{M}}, R_{\mathbf{M}} \upharpoonright S_{\mathbf{M}})$  satisfy  $S_{\mathbf{M}} = S_0$ .

**Example 6.3** Let  $S_0 = \{a, b\}$  and let  $\Psi(S_0)$  be

$$\Psi(S_0) = \forall x (x = \mathbf{a} \vee x = \mathbf{b}) \wedge \mathbf{a} \neq \mathbf{b} \wedge \exists x \forall y \neg y R x \wedge \neg \mathbf{a} R \mathbf{a} \wedge \neg \mathbf{b} R \mathbf{b}$$

Then  $(S_0, \Psi(S_0))$  has the following extensions:

- $a = b = in$
- $a = in, b = out$
- $a = out, b = in$

**Example 6.4** Following Example 6.1, assume we have 5 possible arguments of which we need to select 2 good ones to discuss and assume  $R$  is as in Figure 12 and that we would prefer to discuss any argument  $a_n$  over  $a_{n+1}$  if we could.<sup>7</sup>

We ask what we can take as  $(S_j, R \upharpoonright S_j)$ ?

1. If  $S_j = \{a_1, a_2\}$ , then because  $a_1 \rightarrow a_2$ , we do not get enough choice.
2. If we take  $S_j = \{a_1, a_2, a_3\}$ , we do not get a proper extension.

<sup>7</sup>That is, we rank  $a_1$  as more worthy of discussion than  $a_2$ ,  $a_2$  as more worthy than  $a_3$  etc.

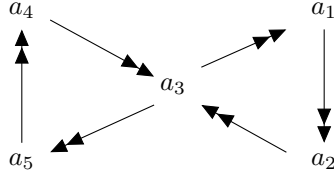


Figure 12:

{541-FAug5}

3. If we take  $S_j = \{a_1, a_2, a_3, a_4\}$  we get the extension  $E = \{a_1, a_4\}$ .

This is our answer.

{541-EAug5}

**Example 6.5 (Disjunctive Networks)** We show that networks with disjunctive attacks (see [19]) can be easily represented as Axiomatic Networks.

Networks with disjunctive attacks have the form  $(S, \rho_\vee)$ , where  $\rho_\vee \subseteq S \times (2^S - \emptyset)$ . A typical disjunctive attack of  $z$  on  $Y = \{y_1, \dots, y_k\}$ ,  $z \rightarrow Y$ , is drawn in Figure 13. One of the options for the semantics for  $(S, \rho_\vee)$  is to understand

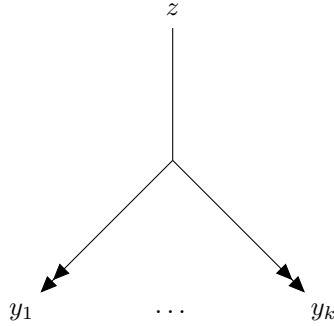


Figure 13:

{541-FAug6}

Figure 13 as saying:

- if  $z = in$  then at least one of  $y_i \in Y$  must be out.

The translation of the above statement for each attack  $z \rightarrow Y$  in  $\rho_\vee$  is  $\Psi(\rho_\vee, R)$ :

$$\Psi(\rho_\vee, R) = \bigwedge_{z \rho_\vee Y} \bigvee_{y \in Y} z R y$$

where  $R$  is a new symbol for a relation  $R \subseteq S \times S$ . Therefore, according to this semantics (called the open reductionist approach in [19]), the extensions of  $(S, \rho_\vee)$  are the same as the extensions of  $(S, \Psi(R))$  as an axiomatic network (see [19, Definition 2.8, Theorem 2.9]).

{541-EAug7}

**Example 6.6 (Conjunctive Networks)** Conjunctive networks have the form  $(S, \rho_\wedge)$ , where  $\rho_\wedge \subseteq (2^S - \emptyset) \times S$ . A typical conjunctive attack of  $Y = \{y_1, \dots, y_k\}$  on  $z$ ,  $Y \rightarrow z$ , is drawn in Figure 14. The condition for conjunctive attacks is



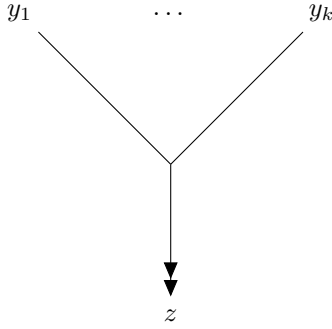


Figure 14:

{541-FAug8}

$$\bigwedge_i y_i = in \text{ implies } z = out.$$

Conjunctive attacks can be implemented using the relation  $R \subseteq S \times S$  of point to point attacks, with the help of additional auxiliary points. For each  $Y \rho_{\wedge} z$ , let the auxiliary points be

$$H(Y, z) = \{\alpha(y, Y, z) \mid y \in Y\} \cup \{\beta(Y, z)\}$$

Consider Figure 15, (this construction is from [20]). Clearly, given  $Y$  and  $z$

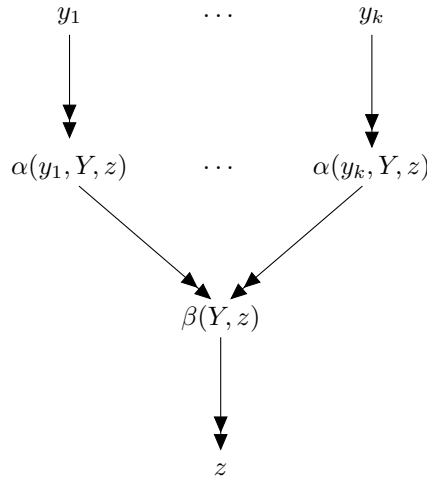


Figure 15:

{541-FAug9}

such that  $Y \rightarrow z$ , if we add the points  $H(Y, z)$  and express that they are all distinct and unique to  $(Y, z)$ , then the point  $\beta(Y, z)$  fills the role of  $Y$  and the attack  $\beta(Y, z) \rightarrow z$  takes the role of  $Y \rho_{\wedge} z$ .

It is also clear that in Figure 15,  $z$  is out only if all  $y \in Y$  are in. So if

we are given a conjunctive network  $(S_0, \rho_0)$ ,<sup>8</sup> with  $\rho_0 \subseteq S_0 \times (2^{S_0} - \emptyset)$  we can pass to a new network  $(S_1, R_1)$  where:  $S_1 = S_0 \cup \bigcup_{Y \rho_0 z} H(Y, z)$ ; and  $R_1$  is defined as in Figure 15 for each  $H(Y, z)$ .  $(S_1, R_1)$  will ‘implement’  $(S_0, \rho_0)$  in the sense that the complete extensions  $E_1$  of  $(S_1, R_1)$  generate, by the projection  $E_0 = E_1 \cap S_0$ , exactly the complete extensions  $E_0$  of  $(S_0, \rho_0)$ . In other words

- $E_0$  is an extension of  $(S_0, \rho_0)$  iff for some extension  $E_1$  of  $(S_1, R_1)$  we have  $E_0 = E_1 \cap S_0$ .

Using the above route we can implement conjunctive networks as axiomatic networks. We need to define a language  $\mathbf{L}$  and a  $\Psi$  in  $\mathbf{L}$  which allows us to identify, using  $\Psi$ , the network  $(S_0, \rho_0)$ . So now let us define  $\rho$  by a formula in an axiomatic argumentation network. The axiom  $\Psi$  we take simply expresses the situation of Figure 15.

We need a language  $\mathbf{L}(S, R, \rho)$  and we define  $\rho$  using  $R$ . We need some auxiliary notation.

Let  $S_0$  be a finite set and assume we have names  $\mathbf{s}$  in  $\mathbf{L}$  for every  $s \in S_0$ . Let  $\mathbf{S}_0 = \{\mathbf{s} \mid s \in S_0\}$ . Given a set of variables  $V$ , let  $D(V)$  be the formula:

$$D(V) = \bigwedge_{\substack{x \neq y \\ x, y \in V}} \neg(x = y) \wedge \forall x \left( \bigvee_{x \in V} z = x \right)$$

Let  $\exists_V$  be the prefix quantifier (for  $V$  finite, i.e.  $V = \{v_1, \dots, v_n\}$ ) then:

$$\exists_V \phi =_{def} \exists v_1 \dots v_n (D(V) \wedge \phi).$$

Now suppose we are given a conjunctive network  $\mathcal{A}_0 = (S_0, \rho_0)$ , with  $\rho_0 \subseteq S_0 \times (2^{S_0} - \emptyset)$ . Let  $\mathbf{L}(S, R, \rho, \mathbf{S}_0)$  be a predicate language with unary  $S$ , binary  $R$  and  $\rho$  and the set of constants  $\mathbf{S}_0 = \{\mathbf{s} \mid s \in S_0\}$ . Let  $V_0$  be the set of distinct variables

$$V_0 = \bigcup_{Y \rho_0 z} \{\alpha(y, Y, z), \beta(Y, z) \mid y \in Y\}.$$

Let  $\Psi(S, R, \rho, \mathbf{S}_0)$ , or  $\Psi$  for short, be the formula

$$\Psi = \exists_{V_0 \cup \mathbf{S}_0} \left[ \begin{array}{l} \forall z \left( S(z) \leftrightarrow \bigvee_{s \in \mathbf{S}_0} z = \mathbf{s} \right) \\ \wedge \bigwedge_{\substack{Y \rho_0 z \\ Y \subseteq S_0 \\ z \in S_0}} \Phi(Y, z) \\ \wedge \bigwedge_{\substack{Y \rho_0 s \\ Y \subseteq S_0 \\ s \in \mathbf{S}_0}} \bigwedge_{y \in Y} \left( y R \alpha(y, Y, s) \wedge \alpha(y, Y, s) R \beta(Y, s) \wedge \beta(Y, s) R s \right) \end{array} \right]$$

<sup>8</sup>We drop the subscript  $\wedge$  to avoid a proliferation of symbols.

where  $\Phi(Y, z)$  is the following wff:

$$\Phi(Y, z) = Y\rho z \leftrightarrow \exists y_1 \dots y_k \left( \bigwedge_{i \neq j} u_i \neq y_j \wedge \bigwedge_j S(y_j) \wedge S(z) \right. \\ \left. \wedge \exists u \exists x_1 \dots x_k \left( \bigwedge_j \neg S(x_j) \wedge \bigwedge_{i \neq j} x_i \neq x_j \wedge \neg S(u) \right. \right. \\ \left. \left. \wedge \bigwedge_j (y_j R x_j \wedge x_j R u) \wedge u R z \right) \right)$$

Let  $\mathbf{M}$  be a model of  $\Psi$ . The original  $(S_0, \rho_0)$  can be obtained from this model by letting  $S_0$  be the extension  $S_{\mathbf{M}}$  of  $S$  in the model. Also the relation  $Y\rho_0 s$ , for  $Y \subseteq S_0$  and  $s \in S_0$  can be defined as the extension  $S_{\mathbf{M}}$  of  $\rho$ .

{541-EAug10}

**Example 6.7 (Abstract Dialectical Frameworks)** We now implement Abstract Dialectical Frameworks (ADF), [21], in our Axiomatic approach (AAF).

We first describe the first order version of ADF. Consider Figure 16. This figure describes the basic configuration of an attack on a point  $x$  in  $(S, R)$ .  $\{y_1, \dots, y_k\}$  are all the attackers of  $x$ . In traditional Dung semantics we have

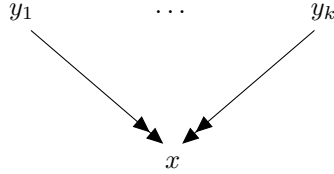


Figure 16:

{541-FAug11}

that:

- $x$  is in iff all  $y_i$  are out

the ADF semantics gives a local condition  $F(x)$  on  $\{y_1, \dots, y_k\}$ , saying

- $x$  is in iff one of certain in/out distributions on  $\{y_1, \dots, y_k\}$  holds.

Assuming that any of these distribution can be expressed in classical propositional logic on  $\{y_1, \dots, y_k\}$ , we can write  $F(x)$  as a wff of the form

$$F(x) = \bigvee_{\epsilon \in \Delta_x} \bigwedge_i y_i = \epsilon(i)$$

where  $\Delta_x$  is a set of vectors  $\epsilon : \{1, \dots, k\} \mapsto \{0, 1\}$  where “ $y = 1$ ” means “ $y$  is in” and “ $y = 0$ ” means “ $y$  is out”.

We can also write  $F(x)$  as  $\bigvee_{\epsilon \in \Delta_x} \bigwedge_i y_i^{\epsilon(i)}$ , where  $y^1 = y$  and  $y^0 = \neg y$ . So we have:

- $x$  is in iff  $\bigvee_{\epsilon \in \Delta_x} \bigwedge_i y_i = \epsilon(i)$

or

- $F_S = \bigwedge_{x \in S} (x \leftrightarrow F(x)) = \top$

The latter formulation of  $F$  is mentioned as formula (1) on the third page of [20]. To conform with the notation of the present paper we should write  $F$  as  $F'$

- $F'(x) = \bigvee_{\epsilon \in \Delta_x} \bigwedge_i In(y_i)^{\epsilon(i)}$ ,

and the full wff is

- $F'_S = \bigwedge_{x \in S} (In(x) \leftrightarrow F'(x)) = \top$ .

The reader should note that the formula  $F'_S$  cannot be considered an argument in our sense of Section 5, because  $F'$  is not a wff about  $R$ . We cannot add it to the specification of the model because it contradicts the traditional *Dung* requirement:

$$\bigwedge_{x \in S} (In(x) \leftrightarrow \bigwedge_{y R x} \neg In(y))$$

which is part of our specification. So our only recourse, if we want to implement (ADF) in (AAF), is to add auxiliary points and follow similar route as taken in Example 6.6.

To simplify the presentation and to save on complicated wffs let us present what is to be done for the configuration of Figure 17. Let  $Y = \{a, b, c\}$  be the

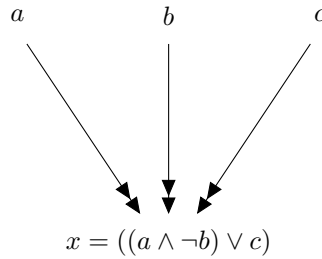


Figure 17:

{541-FAug12}

set of all attackers of  $x$ . The condition for  $x$  to be “in” is that either both  $a = in$  and  $b = out$  or else  $c = in$ .

To implement this figure we add the auxiliary points  $\beta(x, Y)$ ,  $\gamma(x, a, b, Y)$  and  $\gamma(x, c, Y)$ .  $\beta(x, Y)$  stands for  $\neg x$  and  $\gamma(x, a, b, Y)$  stands for the disjunct  $a \wedge \neg b$  and  $\gamma(x, c, Y)$  stands for the disjunct  $c$ .

We also need, for each  $\neg z$  at each disjunct, a point to stand for  $z$ . In our case we have the disjunct  $a \wedge \neg b$  and  $\neg z = \neg b$ . So we need a point  $\delta(\gamma(x, a, b, Y), b)$ . We now create Figure 18.

We will not need some of these additional points, as you will see in the Figure 18 we need only deal with disjuncts which have negated atoms in them.

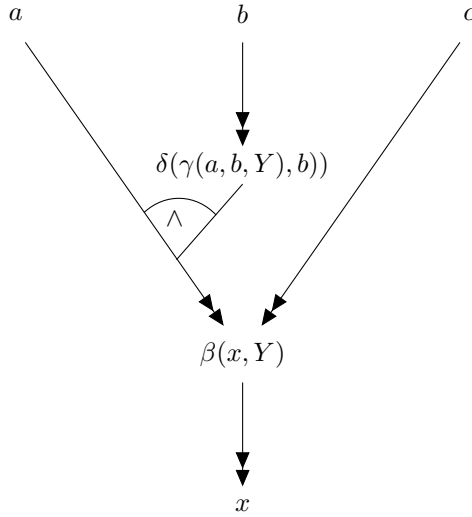


Figure 18:

{541-FAug13}

## 7 Conclusion and discussion

We have translated argumentation networks of the form  $\mathcal{A} = (S, R)$  into intuitionistic logic,  $\tau : A \mapsto \tau A$ , in such a way that the following holds:

1. Complete extensions  $E$  of  $\mathcal{A}$  correspond to models  $h_E$  of  $\tau(\mathcal{A})$  and vice versa.
2. The attack relation in  $\mathcal{A}$ ,  $x \rightarrow y$ , is interpreted intuitionistically as  $x \rightarrow \neg y$  (where  $\rightarrow$  and  $\neg$  are intuitionistic propositional connectives).
3. As a byproduct of this representation we obtained a coherent way of adding, to any  $\mathcal{A}$ , statements  $\phi(R)$  about  $R \in \mathcal{A}$  to serve also as arguments. Moreover, our intuitionistic interpretation  $\tau$  allowed us to give these statements semantics (see Examples 5.1 and 5.2).
4. The reader might have some doubts about the idea of such an interpretation. Let us make several additional points.
  - (a) Members of the argumentation community take pride in the added value of the argumentation point of view. They stress the programme of interpreting, for example, non-monotonic systems into it. They should note that any support or objections to interpreting Abstract Argumentation *out* into systems  $Y$ , may equally be applied to interpreting any system  $X$  *into* an argumentation framework.
  - (b) Whenever we give a good interpretation of a system  $X$  into a system  $Y$ , our understanding of both systems stands to benefit. System  $X$

may get additional meaning as part of  $Y$  and the formal understanding of system  $Y$  may be enhanced through its hosting of system  $X$ . One such example was the translation of modal logic into classical logic via the translation  $\tau$ :

$$\tau(x \vDash \Box A) = \forall y(xRy \rightarrow \tau(y \vDash A))$$

Through this translation, propositional modal logic obtained another semantics, and formal facts about fragments of first order classical logic became apparent. For example, through the observation (e.g. in [24]) that modal logic needs to use only 2 bounded, i.e. guarded, variables in this translation, we obtain decidability results for guarded quantifiers in classical first order logic, (see [25, 26]).

5. We would like to compare the translation of this paper with that of [22].

The paper of Dvorak *et. al.*, [22], is an example of a general translation. The formal mathematical language is second order monadic first order logic. This can serve as the modelling language for the majority, if not all, of the varieties of argumentation networks. It is intended by the authors to be to argumentation as ALGOL is to algorithms. It is an exact mathematical logic language strong enough to express whatever you want to say about argumentation networks. To quote the authors of [22]:

Begin quote.

We propose the formalism of monadic second order logic (MSO) as a unifying framework for representing and reasoning with various semantics of abstract argumentation. We express a wide range of semantics within the proposed framework, including the standard semantics due to Dung, semi-stable, stage, cf2, and resolution-based semantics. We provide building blocks which make it easy and straightforward to express further semantics and reasoning tasks. Our results show that MSO can serve as a *lingua franca* for abstract argumentation that directly yields to complexity results. In particular, we obtain that for argumentation frameworks with certain structural properties the main computational problems with respect to MSO-expressible semantics can all be solved in linear time. Furthermore, we provide a novel characterisation of resolution-based grounded semantics.

End quote.

The monadic second order theory has a symbol  $R$  for the attack relation  $xRy$  and quantifiers over subsets of  $S \in \mathcal{A}$ . For all its strength, it cannot express attacks on attacks. For networks with arguments of the form  $\phi(R)$  the language cannot express, e.g.  $xR\phi(R)$  (i.e. that a point  $x$  attacks the fact that the attack relation  $R$  has the property described by  $\phi$ ). One could extend the language of [22] to allow substitution of formulas like

$\phi(R)$  for variables such as  $y$  in  $xRy$ . But doing this requires an entirely new theory of self-fibring (the general complexities of which are studied in [23]). The solution we can adopt on the basis of the present paper is simple: we use an intuitionistic monadic second order theory. In other words, we combine our intuitionistic interpretation with that of [22]. This simple move also shows the value of the intuitionistic interpretation.

## Acknowledgements

We are grateful to David Pearce for his valuable comments.

## References

- [1] D. Gabbay and M. Gabbay. The attack as strong negation. Part 1. To appear in *Logic Journal of the IGPL*, Manuscript ID: jzv033.
- [2] M. Caminada and D. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2–3), 109–145, 2009.
- [3] David Pearce, Equilibrium logic. *Annals of Mathematics and Artificial Intelligence* June 2006, Volume 47, Issue 1-2, pp 3-41 Date: 14 Sep 2006
- [4] Łukasiewicz, J. Die Logik und das Grundlagenproblem. *Les Entretiens de Zurich sur les Fondaments et la Methode des Sciences Mathematiques*, 69 (1938), 12:82100.
- [5] Gemma Robles. A Simple Henkin-Style Completeness Proof for Gödel 3-Valued Logic  $G_3$ . *Logic and Logical Philosophy*, Volume 23 (2014), 371-390, DOI: 10.12775/LLP.2014.001
- [6] Arend Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie von Wissenschaften, Physikalisch-mathematische Klasse*, pages 42-56, 1930.
- [7] T. Hosoi. The axiomatization of the intermediate propositional systems  $S_n$  of Gödel. *Journal of the Faculty of Science of the University of Tokyo*, 13:183-187, 1966.
- [8] László Kalmár. Zurückführung des Entscheidungsproblems auf den Fall von Formeln mit einer einzigen, bindenden, Funktionsvariablen. *Compositio Mathematica*, 4:137-144, 1936.
- [9] Toshio Umezawa. On intermediate many-valued logics. *Journal of the Mathematical Society of Japan*, 11(2):116-128, 1959.
- [10] D. M. Gabbay. Semantics for higher level attacks in extended argumentation frames. Part 1: Overview. *Studia Logica*, 93(2-3):355-379, 2009.

- [11] Do Duc Hanh, Phan Minh Dung and Phan Minh Thang. Inductive Defense for Modgils Extended Argumentation Framework. *J Logic Computation*, doi: 10.1093/logcom/exq018. First published online: June 16, 2010
- [12] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9-10): 901-934, 2009.
- [13] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Encompassing attacks to attacks in abstract argumentation frameworks. In *ECSQARU*, pages 83-94, 2009.
- [14] Howard Barringer, Dov M. Gabbay, and John Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. In *Mechanizing Mathematical Reasoning*, pages 59-98, 2005.
- [15] D. Gabbay. Theory of Semi-Instantiation in Abstract Argumentation. *To appear in Logica Universalis*, <http://arxiv.org/abs/1504.07020>
- [16] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument and Computation*, 5:1, 3162, 2014. Special Issue: Tutorials on Structured Argumentation, DOI:10.1080/19462166.2013.869766
- [17] Sanjay Modgil. Revisiting Abstract Argumentation Frameworks Theory and Applications of Formal Argumentation, *Lecture Notes in Computer Science*, Volume 8306, 2014, pp 1-15
- [18] Henry Prakken. An abstract framework for argumentation with structured arguments, *Argument and Computation*, 1:2, 93-124, 2010. DOI: 10.1080/19462160903564592
- [19] D. Gabbay and M.Gabbay. Theory of disjunctive attacks, Part 1, To appear on *Logic Journal of IGPL*, DOI 10.1093/jigpal/jzv032
- [20] D. M. Gabbay. Fibring argumentation frames. *Studia Logica*, 93(2-3):231-295, 2009
- [21] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Proc. of the 20th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 102-111, 2010.
- [22] W. Dvorak, S. Szeider, and S. Woltran. Abstract Argumentation via Monadic Second Order Logic, in E. Hullermeier et al. (Eds.): *SUM 2012*, LNAI 7520, pp. 85-98, 2012. Springer-Verlag, Berlin Heidelberg, 2012.
- [23] D. Gabbay, *Fibring Logics (Monograph)*, Oxford University Press, 1998, 600 pp.
- [24] D. Gabbay. Expressive functional completeness in tense logic. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 91-117. D. Reidel Publishing Company, 1981.



- [25] Carlos Areces, Christof Monz, Hans de Nivelle, and Maarten de Rijke. The Guarded Fragment: Ins and Outs. arecesmonzdenivellederijke.pdf
- [26] H. Andreka, J. van Benthem, and I. Nemeti, Modal languages and bounded fragments of predicate logic, *Journal of Philosophical Logic*, 27(3):217-274, 1998.

## Appendix

### A The logic $\mathbf{G}_3$

The logic is called  $\mathbf{G}_3$  because it corresponds to Gödel's  $n$ -valued logic with  $n = 3$ . But the logic is due to Heyting. It was first axiomatised by Lukasiewicz in 1938 by

$$(\neg x \rightarrow y) \rightarrow (((y \rightarrow x) \rightarrow y) \rightarrow y)$$

But there is shorter axiom due to Umezawa later proved complete by Hosoi:

$$x \vee (\neg y \vee (x \rightarrow y))$$

David Pearce [3], calls the logic *HT* or 'here-and-there' because of the two linearly ordered worlds.

$\mathbf{G}_3$  can also be characterised by adding the following additional two axioms to the axioms of propositional intuitionistic logic.

**Dummett LC axiom.**

$$(x \rightarrow y) \vee (y \rightarrow x)$$

This gives linearity.

**Peirce's axiom for height 2.**

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

### B Semantics for $\mathbf{G}_3$

We have two possible worlds  $t$  and  $s$ , with the ordering  $t < s$ .  $t$  is considered the actual world. The language of  $\mathbf{G}_3$  contains atoms and the intuitionistic connectives  $\{\wedge, \vee, \rightarrow, \neg, \perp, \top\}$ .

An assignment  $h$  to the atoms gives them values in  $t$  and in  $s$ . We write  $h(x, q) \in \{\top, \perp\}$  for  $x \in \{t, s\}$  and  $q$  atomic. We also write  $h(q) = (v_1, v_2)$  where  $v_1 = h(t, q)$  and  $v_2 = h(s, q)$ . We require that

$$h(t, q) = \top \text{ implies } h(s, q) = \top,$$

so the value  $h(q) = (\top, \perp)$  is forbidden.

We evaluate wff of the language as follows. For a given  $h$ , and a given  $a \in \{t, s\}$

{sem-prop}

- $a \models_h \top$
- $a \not\models_h \perp$
- $a \models_h q$  iff  $h(a, q) = \top$  (for any atomic  $q$ )
- $a \models_h A \wedge B$  iff  $a \models_h A$  and  $a \models_h B$
- $a \models_h A \vee B$  iff  $a \models_h A$  or  $a \models_h B$
- $a \models_h \neg A$  iff  $b \not\models_h A$  for all  $b \in \{t, s\}$  s.t.  $a \leq b$ .
- $a \models_h A \rightarrow B$  iff  $b \not\models_h A$  or  $b \models_h B$ , for all  $b \in \{t, s\}$  s.t.  $a \leq b$ .

We have the following theorems:

1. For any  $A$ , if  $t \models_h A$  then  $s \models_h A$ .
2.  $\mathbf{G}_3 \vdash A$  iff for any  $h$ ,  $t \models_h A$ .

## C Semantics for predicate $\mathbf{G}_3$

{sem-quant}

We consider a special version of predicate  $\mathbf{G}_3$ , where the language has the additional quantifiers  $\forall x$ ,  $\exists x$  and the predicates  $xRy$  and  $In(x)$ . The logic is a constant domain logic for the worlds  $\{t, s\}$  and the relation  $R$  is *decided*, meaning  $xRy$  holds at both  $t$  and  $s$  or neither, for any  $x$  and  $y$ .

We thus extend the semantics of Section B with a domain  $D$ , relation  $\rho \subseteq D \times D$ , and a relation  $\iota \subseteq \{t, s\} \times D$  where  $\iota(t, d) \subseteq \iota(s, d)$  for  $d \in D$ . Assignments, which we now denote by  $v$ , are functions from variable symbols to elements of  $D$ .

We evaluate wff of the language as follows. For a given  $v$ , and a given  $a \in \{t, s\}$

- $a \models_v \top$
- $a \not\models_v \perp$
- $a \models_v xRy$  iff  $v(x)\rho v(y)$
- $a \models_v In(x)$  iff  $\iota(a, v(x))$
- $a \models_v A \wedge B$  iff  $a \models_v A$  and  $a \models_v B$
- $a \models_v A \vee B$  iff  $a \models_v A$  or  $a \models_v B$
- $a \models_v \neg A$  iff  $b \not\models_v A$  for all  $b \in \{t, s\}$  s.t.  $a \leq b$ .
- $a \models_v A \rightarrow B$  iff  $b \not\models_v A$  or  $b \models_v B$ , for all  $b \in \{t, s\}$  s.t.  $a \leq b$ .
- $a \models_v \forall x A$  iff  $b \models_{v[x \mapsto d]} A$  for all  $d \in D$  and for all  $b \in \{t, s\}$  s.t.  $a \leq b$ .
- $a \models_v \exists x A$  iff  $a \models_{v[x \mapsto d]} A$  for some  $d \in D$ .<sup>9</sup>

<sup>9</sup>Where  $v[x \mapsto d]$  is a function identical to  $v$  except  $v(x) = d$ .