

# Belief Revision in non-classical logic I

Dov Gabbay and Odinaldo Rodrigues

Department of Computer Science,  
King's College, London,  
WC2R 2LS, United Kingdom  
{dg,odinaldo}@dcs.kcl.ac.uk

Alessandra Russo

Department of Computing,  
Imperial College,  
180's Queen's Gate,  
London SW7 2BZ, United Kingdom  
ar3@doc.ic.ac.uk

## Abstract

In this paper, we propose an unifying belief revision approach for families of (non-classical) logics whose semantics are first-order axiomatisable. Given any such (non-classical) logic  $\mathcal{L}$ , the approach enables the definition of belief revision operators for  $\mathcal{L}$ , in terms of a classical AGM belief revision operation. The approach is illustrated by considering Belnap's four-valued logic. We analyse in what circumstances the AGM properties are preserved and discuss limitations and advantages of the approach from both theoretical and practical viewpoints.

## 1 Introduction

This paper presents a method for revising theories of non-classical logics. The general idea is to define a belief revision operator for these logics in terms of a standard belief revision operator for classical logic. Typically, given a classical logic theory  $\Delta$  and an input formula  $\psi$ , a revision process  $*_a$  gives us a new theory  $\Gamma = \Delta *_a \psi$ , corresponding to the revision of  $\Delta$  by  $\psi$  and satisfying some desirable properties (e.g. the AGM postulates for belief revision [Gärdenfors, 1988]). Our approach shows that this machinery can be *exported* to other (non-classical) logics, whenever their semantics is first-order logic axiomatisable.

This is done by translating the mechanics of a given object logic  $L$  into classical logic, performing the revision process in classical logic and then translating the results back into the object logic. Let  $\tau$  denote a translation function from  $L$  into classical logic,  $\mathcal{A}_L$  be a sound and complete classical logic axiomatisation of the mechanics of  $L$  with respect to its entailment notion, and  $Acc$  be a first-order characterisation of *acceptable*  $L$ -theories. We define a revision operator  $*_L$  in the logic  $L$  as follows:

$$\Delta *_L \psi = \{ \beta \mid \Delta^\tau *_a (\psi^\tau \wedge \mathcal{A}_L \wedge Acc) \vdash \beta^\tau \} \quad (1)$$

The motivation for this definition is as follows.  $\Delta^\tau$  is the translation of the  $L$ -theory  $\Delta$  to classical logic.  $\Delta$  is to be revised by  $\psi$ , which in classical logic is translated as  $\psi^\tau$ . The revision of  $\Delta$  by  $\psi$  is defined in terms of the classical revision of  $\Delta^\tau$  by  $\psi^\tau$ . The inclusion of the axiomatisation  $\mathcal{A}_L$  allows the semantic properties of the object logic  $L$  to be preserved

during the revision process, so that the revised theory can be mapped back to  $L$ . This is why we have defined the revision of  $\Delta^\tau$  with respect to  $\psi^\tau \wedge \mathcal{A}_L$ .

Notice that we have also included the formula  $(Acc)$  in the revising part of (1). For non-classical logics which are extensions of classical logic, such as modal logics, the notion of inconsistency is still classical, so it is identical to the notion of inconsistency in the target (classical) logic. If a given object formula  $\psi$  is inconsistent with an object theory  $\Delta$ , then  $\psi^\tau$  would also be inconsistent (in classical logic) with  $\Delta^\tau$ . Hence the classical revision would be able to revise the (translated) theory and provide a revised theory that is also consistent in the object logic. For these types of non-classical logics, the revision operator  $*_L$  could be defined more simply as follows:

$$\Delta *_L \psi = \{ \beta \mid \Delta^\tau *_a (\psi^\tau \wedge \mathcal{A}_L) \vdash \beta^\tau \}$$

This is the approach adopted and discussed in [Gabbay *et al.*, 2000]. However, even in these cases, some domain-dependent notions of acceptability could still be provided to further refine the revision process, as it is done, for instance, with integrity constraints in database updates. The formula  $Acc$  would, in this case, be the (classical logic) definition of domain-dependent constraints, and the revision operator  $*_L$  to be defined as in (1).

In the case of non-classical logics that are not extensions of classical logic, such as Belnap's logic, the notion of consistency differs from that of classical logic. Object theories may not be necessarily semantically inconsistent. Let  $Cn$  be the closure of a set of formulae under the consequence relation of classical logic and consider the formula  $p \wedge \neg p$ . This formula is not contradictory in Belnap's logic. If a translation to classical logic is sound and complete, then  $\{p^\tau\}$  will be consistent with  $(\neg p)^\tau$ , and consequently, the revision of  $p$  by  $\neg p$  done via classical logic will be simply  $\{p \wedge \neg p\}$ . This might not be what one expects in the object logic. In these circumstances, the revision would be equivalent to an expansion, which is how belief revision has been defined so far for logics such as paraconsistent logics [Restall and Slaney, 1995].

What we propose in this paper is a different approach to revision for non-classical logics (including paraconsistent logics). Our view is that a revision process for such logics would benefit from a shift towards a notion of *acceptable belief sets*, by imposing specific restrictions on what an acceptable theory should be in the object logic  $L$ , and therefore on what

is the effect of revising a theory with respect to a notion of acceptability. We assume that such notion can be provided either by the specific domain of applications, or by logical properties, with the aim of investigating the relationship between these properties and the underlying classical AGM postulates.

## 2 Belief Revision in Classical Logic

The term *belief revision* is generally used to describe the process of changing a given logical theory to accommodate new (possibly contradictory) information, in such a way that the process will satisfy some desirable properties. One important feature is that one wants the change to be minimal in some sense. Notice also that this is intrinsically a non-monotonic form of reasoning.

In this paper we concentrate only on the standard approach to belief revision. By this we mean, belief revision operations satisfying the AGM postulates [Gärdenfors, 1988]. Therefore, if  $K$  is a belief set, then we assume that  $K = \text{Cn}(K)$ . Similarly, if  $\varphi \in K$ , we say that  $\varphi$  is *accepted* in  $K$ .

The general task of the revision process can be seen as determining what is rational to believe when a new belief (possibly) contradicting the belief in the current belief set is accepted. The AGM postulates have often been considered as standard criteria for evaluating revision operations and provide, in a sense, a *minimal* set of conditions to be satisfied by such operations. We will use them to investigate the properties of the operator obtained in our approach.

The AGM postulates for belief revision, as given in [Gärdenfors, 1988], can be found below. In the following presentation,  $K$  is a belief set and  $A$  and  $B$  are formulae. The symbol  $K_{\perp}$  is used to denote the inconsistent belief set (the set of all formulae in the language).

### AGM postulates for Belief Revision

- (K\*1)  $K *_a A$  is a belief set
- (K\*2)  $A \in K *_a A$
- (K\*3)  $K *_a A \subseteq \text{Cn}(K \cup \{A\})$
- (K\*4) If  $\neg A \notin K$ , then  $\text{Cn}(K \cup \{A\}) \subseteq K *_a A$
- (K\*5)  $K *_a A = K_{\perp}$  only if  $A$  is contradictory
- (K\*6) If  $A \equiv B$ , then  $K *_a A \equiv K *_a B$
- (K\*7)  $K *_a (A \wedge B) \subseteq \text{Cn}((K *_a A) \cup \{B\})$
- (K\*8) If  $\neg B \notin K *_a A$ , then  $\text{Cn}(K *_a A \cup \{B\}) \subseteq K *_a (A \wedge B)$

Postulate (K\*1) requires that the result of the revision operation also be a belief set. (K\*2), known as the *success postulate*, requires that the new information is accepted in the revised belief set. Postulates (K\*3) and (K\*4) define in a sense the “lower” and “upper” bounds of the revision operation, respectively. In [Gabbay *et al.*, 2000], we have argued that together they amount to the following principle:

- (K\*<sub>3,4</sub>) If  $A$  is consistent with  $K$ , then  $K *_a A = \text{Cn}(K \cup \{A\})$ .

(K\*5), instead, only requires  $K$  to be consistent whenever  $A$  is consistent. When we check the AGM postulates for logics other than classical, we may have a different notion of consistency and so we are free to interpret what

we want the revision to do in the case of inconsistency according to what is reasonable in the object (non-classical) logic. AGM for classical logic gives us no clue beyond (K\*5) as to what to require when  $(K *_a A) \cup \{B\}$  is inconsistent. (K\*6) specifies that the revision process should be independent of the syntactic form of the sentences involved. Finally, postulates (K\*7)–(K\*8) have to do with the relationship of inputting  $(A, B)$  as a sequence (first revising by  $A$ , then expanding by  $B$ ), as compared with revising by  $\{A, B\}$  at the same time (i.e. revising by  $A \wedge B$ ). It is well known that AGM does not say enough about sequences of revisions and their properties. These postulates are the bare minimum (see, for instance, [Darwiche and Pearl, 1996; 1997]). In fact, (K\*7)–(K\*8) do not tell us anything new (beyond what we can deduce from earlier postulates), except in the case where  $B$  is consistent with  $K *_a A$ . In [Gabbay *et al.*, 2000], we have argued that (K\*7) and (K\*8) together are equivalent to the following postulate:

- (K\*<sub>7,8</sub>)  $\text{Cn}((K *_a A) \cup \{B\}) = K *_a (A \wedge B)$ , when  $B$  is consistent with  $K *_a A$

For non-classical logics, we are committed only to (K\*<sub>7,8</sub>). Other cases involving inconsistency can have properties dictated by the object logic requirements.

## 3 Belief Revision in non-Classical Logics

In this section, we describe our approach for defining belief revision operators for non-classical logics. This is based on the following components: *i*) a sound and complete classical logic axiomatisation of the semantics of the object logic  $L$ , *ii*) a domain-dependent notion of “acceptability” for theories of  $L$  and *iii*) a classical AGM belief revision operation.

In general, different translation mechanisms can be defined from a given object logic to classical logic, depending on the underlying semantic structures used by the object logic. In the case of non-classical logics that are extensions of classical logic, the translation mechanism essentially maps the notion of satisfiability of atomic formulae to associated first-order predicates with one or more (additional) arguments to encapsulate elements of the underlying structure of the semantics. For instance, in the case of propositional modal logic, the semantic notion  $w \Vdash p$ , for a possible world  $w$  and atomic formula  $p$  can be associated with a first-order formula  $P(w)$ , and analogously,  $w \not\Vdash p$ , with the formula  $\neg P(w)$ . Appropriate axioms may have to be defined to express the correct behaviour of the connectives (and indeed semantics) of the object logic, for instance, to represent special conditions of the accessibility relation and connectives that do not exist in the target language. Examples of this translation mechanism can be found in [Ohlbach, 1991].

For non-classical logics, whose semantics are significantly different from that of classical logic, the translation mechanism has to mirror the structures upon which the semantics as well as the notion of entailment are based. In the case of Belnap’s four-valued logic, the semantical notion of entailment is based on a lattice with four elements, called the *logical lattice* (**L4**). The translation mechanism needs to reflect the structure of the lattice and express (declaratively) the different truth-value assignments that a formula can assume under

given valuations. An axiomatisation expressing the behaviour of the connectives needs also to be defined. Independently of the translation mechanism adopted, such axiomatisation has to be *sound* and *complete* with respect to the semantic notion of entailment of the object logic  $L$ . This ensures that, given a translation function  $\tau$  and a classical logic axiomatisation  $\mathcal{A}_L$ , for any given theory  $\Delta$  and formula  $\alpha$  in  $L$ ,

$$\Delta \vdash_L \alpha \text{ iff } \mathcal{A}_L \cup \Delta^\tau \vdash \alpha^\tau \quad (2)$$

where  $\Delta^\tau$  and  $\alpha^\tau$  are the classical translations of  $\Delta$  and  $\alpha$  respectively.<sup>1</sup> Throughout the paper, we refer to  $\mathcal{A}_L$  as a sound and complete axiomatisation of  $L$  in classical logic. Our method is thus applicable only to logics whose semantics are first-order logic axiomatisable.

The second component of our approach is a domain-dependent notion of *acceptability* for a theory in  $L$ . Given a revision operator  $*_a$  for classical logic, we could define a revision operator  $*_L$  for the logic  $L$  in the following way.

$$\Delta *_L \psi = \{ \beta \mid \Delta^\tau *_a (\psi^\tau \wedge \mathcal{A}_L) \vdash \beta^\tau \} \quad (3)$$

However, the notion of consistency in the object logic may differ from that of classical logic. The AGM revision relies somehow on the notion of (classical) consistency. This can be easily understood by analysing postulate (K<sub>3,4</sub><sup>\*</sup>). Revision is only triggered when the new information is inconsistent with the current belief set. In this sense, revision in the translation can only be performed if the translated theory turns out to be *classically* inconsistent. The direct application of our method to logics such as Belnap’s four-valued logic, which does not have any notion of inconsistency, would therefore yield simply to an expansion, since the translation of any Belnap theory is always classically consistent. This is a consequence of the fact that AGM belief revision is too dependent on the mechanics of classical logic itself.

Our view is that belief revision processes would benefit from a shift towards the more general notion of *acceptable* belief sets. In the particular case of classical logic, it makes sense to require that an acceptable belief set be *at least* consistent, since inconsistency trivialises the notion of consequence in classical logic. However, there is no reason why more specific restrictions should not be imposed on what an acceptable belief set is in classical logic, as it is done for instance, with integrity constraints in database updates. This paves the way for the definition of *relative* revision procedures, where the process is guided by a fixed theory representing a particular notion of acceptable belief sets. For the purposes of this paper, our relative revision is defined in terms of a notion of acceptability explicitly included in the revision process. Our definition of belief revision for a non-classical logic  $L$  is therefore as follows.

**Definition 1** [Belief revision in  $L$ ] Let  $*_a$  be a revision operator for classical logic and  $Acc$  a formula expressing a (domain-dependent) notion of acceptability of  $L$ -theories. The revision of  $\Delta$  by  $\psi$ ,  $\Delta *_L \psi$ , is defined as

$$\Delta *_L \psi = \{ \beta \mid \Delta^\tau *_a (\psi^\tau \wedge \mathcal{A}_L \wedge Acc) \vdash \beta^\tau \} \quad \square$$

<sup>1</sup>We use the symbol  $\vdash$  without subscript to denote entailment in classical logic and with subscript to denote entailment on a object logic.

Hence, a theory  $\Delta$  can be revised by a formula  $\psi$ , by instead revising classically  $\Delta^\tau$  by  $\psi^\tau$ . The semantic properties of the object logic  $L$  and the notion of acceptability are preserved by performing the classical revision process also with respect to the axiomatisation  $\mathcal{A}_L$  and the formula  $Acc$ , to guarantee that the resulting revised theory is an acceptable theory in the object logic  $L$ .

## 4 Belnap’s four-valued logic

Standard familiar systems such as classical logic, modal logic and intuitionistic logic have in common the principle that contradicting information entails any arbitrary sentence. This principle, known as *ex falsum quod libet*, is, however, not always appropriate to describe real application deduction processes, where information is often deduced from quite possibly inconsistent databases. Alternative systems have been developed, examples of which include the logic of *first-degree entailment* (also known as system **E**) and the *relevant implication* system (or system **R**), in which deductions between formulae hold only when there is some “connection” between the formulae (e.g. the formulae share some sentential variable). In [Belnap, 1977b], Belnap provides a semantic characterization of first-degree entailment together with a sound and complete axiomatisation, emphasising its connection with the problem of “how a computer should think” [Belnap, 1977a].

We provide a translation of Belnap’s semantics into a set of first-order logic formulae. Sets of Belnap formulae are translated into a conjunction of atomic predicates. An appropriate classical axiomatisation is defined, which captures the semantic behaviour of Belnap connectives, thus allowing Belnap’s notion of entailment to be expressed in terms of classical entailment from the translated theories. This embedding into classical logic has two main advantages. The first one is to provide the basis for analysing belief revision operations for these types of logics. Such an analysis for propositional modal logic can be found in [Gabbay *et al.*, 2000]. Secondly, theorem proving for four-valued logic can be done by using existing theorem provers for classical logic on the classical logic translation of these logics.

### Notation & Terminology

We reserve lower-case letters for atomic symbols in Belnap’s language. Greek-letters will be used in general to refer to wffs of the Belnap language, i.e. the “object logic”. Larger entities such as structures, sets, theories and languages will often be typeset in caligraphic font, e.g.,  $\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$ . In this section and the next, all formulae, sets of formulae, valuations, etc, refer to objects of Belnap’s language defined below.

### 4.1 Entailment in Belnap’s four-valued logic

In [Belnap, 1977a], Belnap describes his semantic characterisation of four-valued logics as an appropriate logic for expressing *practical* deductive processes. In database management or question-answer systems, collections of data are prone to include either explicit or hidden inconsistencies. The motivation for Belnap’s approach is to provide a logic less sensitive to the problem of inconsistency.

**Syntax** Let  $\mathcal{L}_B$  be the Belnap language composed of a countable set of propositional letters  $\mathcal{P}_B = \{p, q, r, \dots\}$  and the connectives  $\neg, \wedge$  and  $\vee$ . The set of wffs is obtained by the standard construction mechanism of formulae. For the finite case, a Belnap theory can be seen as a single formula given by the conjunction of all wffs in the set. The formula  $\neg p \wedge (\neg q \vee r) \wedge \neg r$  is an example of a finite Belnap theory.

In proof theoretical terms, Belnap's four-valued logic is characterised by a finite axiomatization. Given two Belnap wffs  $\alpha$  and  $\beta$ , the expression  $\alpha \rightarrow \beta$  denotes that  $\alpha$  entails  $\beta$ . In this sense, the symbol  $\rightarrow$  can be seen as a derivability relation between formulae, or equally between theories and a formula. For space limitations, we cannot present the axiomatization here, please refer to [Belnap, 1977b] for a comprehensive presentation. We will concentrate instead on the logic's semantic on which our translation is based. In this logic, the semantic and proof theoretic notions of derivability are equivalent, as will be shown later.

**Semantics** The semantics underlying Belnap's logic is four-valued. Let  $\mathbf{4}$  be the set  $\{T, F, \text{Both}, \text{None}\}$ . The elements of this set are the four different truth-values which an atomic sentence can have within a given "state of information". The intuitive meaning of these values is given as follows:

1.  $p$  is stated to be true only (T)
2.  $p$  is stated to be false only (F)
3.  $p$  is stated to be both true and false (Both), and
4.  $p$ 's status is unknown. That is, none of the above (None).

The four values form a lattice, called the *approximation lattice* and denoted by  $\mathbf{A4}$  where the ordering relation  $\sqsubseteq$  goes "uphill" and respects the monotonicity property, in the sense that information about the truth-value of a formula "grows" from None to Both.  $\mathbf{A4}$  can be seen in Figure 1.

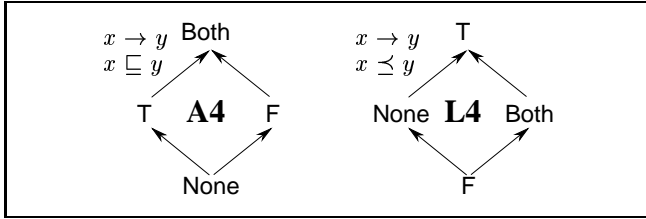


Figure 1: The approximation ( $\mathbf{A4}$ ) and logical ( $\mathbf{L4}$ ) lattices.

The interpretation of the connectives in complex formulae is shown in Figure 2. The truth-tables constitute a lattice, called *logical lattice* and denoted by  $\mathbf{L4}$  (Figure 1). In  $\mathbf{L4}$ , logical conjunction is identified with the meet operation and logical disjunction with the join operation.

The notion of an interpretation of formulae is expressed in Belnap's logic in terms of *set-ups*. A set-up  $s$  is a mapping of the atomic formulae into  $\mathbf{4}$ . Using the truth tables given in Figure 2, each set-up can be extended to a mapping of all formulae into  $\mathbf{4}$ , in the standard inductive way. We call this extended set-up a *4-valuation* and denote it with  $v$ . Thus, for any given Belnap formula  $\alpha$  and set-up  $s$ , the valuation  $v(\alpha)$  is always well-defined. This makes Belnap's semantic somewhat different from the classical semantics, because the no-

$\wedge$	None	F	T	Both
None	None	F	None	F
F	F	F	F	F
T	None	F	T	Both
Both	F	F	Both	Both

$\vee$	None	F	T	Both
None	None	None	T	T
F	None	F	T	Both
T	T	T	T	T
Both	T	Both	T	Both

$\neg$	None	F	T	Both
None	None	T	F	Both

Figure 2: Truth-tables for the connectives in  $\mathcal{L}_B$ .

tion of model, that is, an interpretation that makes a formulae true is non-existent.

The notion of semantic entailment is expressed in terms of a partial ordering  $\preceq$  associated with the logical lattice  $\mathbf{L4}$ . We will denote the semantic entailment relation with  $\Rightarrow$  to distinguish it from the proof theoretic notion of entailment  $\rightarrow$ . The two notions are equivalent, as given by the correspondence 4.

**Definition 2** Let  $\alpha$  and  $\beta$  be formulae. We say that  $\alpha$  entails  $\beta$ , written  $\alpha \Rightarrow \beta$ , if for all  $\mathbf{4}$ -valuations  $v$ ,  $v(\alpha) \preceq v(\beta)$ , where  $\preceq$  is the partial ordering associated with  $\mathbf{L4}$ . Analogously, a *non empty* finite set of formulae  $\Gamma$  entails  $\alpha$ , if the conjunction of all formulae in  $\Gamma$  entails  $\alpha$ .  $\square$

The correspondence below is known to be correct [Belnap, 1977a].

$$\text{(Correspondence)} \quad \alpha \rightarrow \beta \text{ iff } \alpha \Rightarrow \beta \quad (4)$$

Notice that if we restrict our attention to valuations into  $\{F, T\}$  only, we get the familiar classic logic notions. We need a parallel to the notion of consistency, this would be *acceptability*. We say a Belnap theory  $\Delta$  is *acceptable*, if for any wff  $\alpha$ ,  $\Delta$  does not prove both  $\alpha$  and  $\neg\alpha$ . This is our definition and it seems reasonable to us. A consistent/acceptable theory can tell us that a given  $\alpha$  is true (T), false (F) or don't know (None), but if it says that it is both true and false (Both), then something in the knowledge about  $\alpha$  went wrong.

**Definition 3** [Acceptability of Belnap theories] A theory  $\Delta$  is *acceptable*, if  $\{\gamma \mid \Delta \rightarrow \gamma \text{ and } \Delta \rightarrow \neg\gamma\} = \emptyset$ .  $\square$

Note that we do not have that if  $\Delta$  is unacceptable, then it can prove everything. After all, this logic is paraconsistent. We now introduce some terminology which will be used later in the article.

**Definition 4** Let  $\alpha$  be a formula and  $v$  be a  $\mathbf{4}$ -valuation.

- $\alpha$  is *at least true* under  $v$  if  $v(\alpha) = T$  or  $v(\alpha) = \text{Both}$ .
- $\alpha$  is *at least false* under  $v$  if  $v(\alpha) = F$  or  $v(\alpha) = \text{Both}$ .
- $\alpha$  is *not true* under  $v$  if  $v(\alpha) = F$  or  $v(\alpha) = \text{None}$ .
- $\alpha$  is *not false* under  $v$  if  $v(\alpha) = T$  or  $v(\alpha) = \text{None}$ .  $\square$

Using the above terminology, the notion of semantic entailment between a theory and a formula given in Definition 2 can be equivalently expressed as follows.

**Definition 5** Let  $\Gamma$  be a set of formulae and  $\alpha$  a formula.  $\Gamma$  entails  $\alpha$  if and only if for every  $\mathbf{4}$ -valuation  $v$ ,

- i) if all the formulae in  $\Gamma$  are at least true under  $v$ , then  $\alpha$  is at least true under  $v$ ;
- ii) if all the formulae in  $\Gamma$  are not false under  $v$ , then  $\alpha$  is not false under  $v$ .  $\square$

This definition will play an important role in the soundness and completeness proofs of the first-order Belnap translation with respect to Belnap's semantics.

## 5 Generalizing Belnap's notion of entailment to infinite theories

In Belnap's logic the truth-value of a conjunction of formulae is defined inductively from the truth-values of its conjuncts. Alternatively, this value can be calculated by taking the meet of the truth-values of the conjuncts in **L4**. In that sense, finite sets of formulae and conjunctions behave in the same way. However, the notion of meet is more general, because it can be applied to an infinite set of truth-values.

For technical reasons, we need to generalise the notion of entailment to infinite sets of sentences, because when we apply the revision operator in classical logic and translate the results back to Belnap's language, we get an infinite set of Belnap formulae. This is as it should be since by  $(K^*1)$ , the result of the revision is a belief set, and hence closed under the consequence relation.

We therefore now use the more general notion of meet instead of the original inductive definition of  $\wedge$ .

**Definition 6** Let  $\Delta$  be a (possibly infinite) set of formulae,  $S$  be the set of all set-ups and  $s$  an arbitrary set-up. We define  $val : S \times \mathcal{P}(\mathbf{wff}_{\mathcal{L}_B}) \mapsto \mathbf{4}$  as<sup>2</sup>

$$val(s, \Delta) = \sqcap \{s(\varphi) \mid \varphi \in \Delta\} \quad \square$$

We now generalize the notion of entailment in Belnap's logic.

**Definition 7** Let  $\Delta$  be a (possibly infinite) set of formulae and  $\varphi$  a formula.

$$\Delta \rightarrow \varphi \text{ iff } \forall s [val(s, \Delta) \preceq s(\varphi)] \quad \square$$

In the finite case, the above definition corresponds to the original notion of entailment in Belnap's logic:

**Proposition 1** Let  $\Delta$  be a finite set of formulae and  $\delta = \bigwedge_{\beta_i \in \Delta} \beta_i$ .  $val(s, \Delta) = s(\delta)$ .

**Proof** The proof is done by induction on the size of  $\Delta$ .  
*Base case:*  $|\Delta| = 1$ .  $\Delta = \{\delta'\}$ , for some  $\delta'$ .  $val(s, \Delta) = \sqcap \{s(\delta')\}$ . By the definition of  $\sqcap$ ,  $val(s, \Delta) = s(\delta') = s(\delta)$ .  
*Inductive Step:* The proposition holds for any  $\Gamma$  such that  $|\Gamma| \leq k$ . Now, suppose that  $|\Delta| = k + 1$ .  $\Delta = \Gamma \cup \{\beta\}$ , for some  $\Gamma$  of cardinality  $k$ . Let  $\gamma = \bigwedge_{\gamma_i \in \Gamma} \gamma_i$ . It follows that  $val(s, \Delta) = \sqcap \{s(\delta_i) \mid \delta_i \in \Delta\} = val(s, \Gamma) \sqcap s(\beta)$ , by the distributivity property of  $\sqcap$ . By the induction hypothesis,  $val(s, \Gamma) = s(\gamma)$ . Thus,  $val(s, \Delta) = s(\gamma) \sqcap s(\beta) = s(\gamma \wedge \beta) = s(\delta)$ , by associativity and distributivity of  $\wedge$ .  $\square$

<sup>2</sup> $\mathcal{P}(\mathbf{wff}_{\mathcal{L}_B})$  is the power set of the set of all well-formed Belnap formulae.

**Corollary 1** Let  $\Delta$  be a finite set of formulae and  $\varphi$  a formula and let  $\delta = \bigwedge_{\beta_i \in \Delta} \beta_i$ .

$$val(s, \Delta) \preceq s(\varphi) \text{ iff } s(\delta) \preceq s(\varphi) \quad \square$$

The extended definition of entailment satisfies the finiteness property, as shown in Theorem 2. The proof of this property uses the following theorem and definitions.

**Definition 8** [Ultra set-up] Let  $I$  be a nonempty set,  $D$  an ultrafilter over  $I$  and  $s_i$  a set-up for  $\mathcal{L}_B$ , for each  $i \in I$ .  $S = \Pi_D s_i$  is a function  $S : \mathcal{P}_B \mapsto \mathbf{4}$  defined as follows. For each  $p \in \mathcal{P}_B$ ,  $S(p) = u$  iff  $\{i \in I \mid s_i(p) = u\} \in D$   $\square$ .

The definition above is extended to complex formulae in the usual way, using Belnap's truth tables. For example,  $S(\alpha \wedge \beta) = S(\alpha) \wedge S(\beta)$ . It is useful to refer to the particular value of  $y$  assigned to formulae by the function  $S$  above. The following theorem shows that an ultra set-up as defined above is indeed a  $\mathbf{4}$ -valuation as defined before.

**Theorem 1 (The fundamental theorem)** For each  $\alpha \in \mathcal{L}_B$ ,  $S(\alpha) = u$  iff  $\{i \in I \mid s_i(\alpha) = u\} \in D$ .

**Proof** The proof is by induction on the complexity of a formula  $\alpha$ .

("Only if") The base case follows directly from Definition 8. Assume that the theorem holds for any formula  $\beta$  with at most  $k$  connectives. *Inductive step:* The proof is done by cases on the main connective of  $\alpha$ .

$(\neg)$   $\alpha$  is of the form  $\neg\beta$  for some  $\beta$ . Let  $S(\neg\beta) = u$ , for some  $u$ . By the definition of  $S$ ,  $S(\neg\beta) = \neg S(\beta) = \neg u_1$ , for some  $u_1 = S(\beta)$ . Notice that, by the inductive hypothesis, the set  $N = \{n \mid s_n(\beta) = u_1\} \in D$ . We show that the set  $M = \{m \mid s_m(\neg\beta) = u\} = N$ , and therefore  $M \in D$ . Let  $k \in M$ . Then  $s_k(\neg\beta) = \neg s_k(\beta) = \neg u_2$ , for some  $u_2$ . By the definition of  $\neg$ ,  $\neg u_1 = \neg u_2$  implies that  $u_1 = u_2$ . Therefore  $k \in N$ . Let  $k \in N$ . Then  $s_k(\beta) = u_1$ . By definition of  $\neg$ ,  $s_k(\neg\beta) = \neg s_k(\beta) = \neg u_1 = u$ . Hence,  $k \in M$ .

$(\wedge)$   $\alpha$  is of the form  $\beta_1 \wedge \beta_2$ , for some  $\beta_1$  and  $\beta_2$ . Let  $S(\alpha) = u$ . By definition of  $S$ ,  $u = S(\beta_1) \wedge S(\beta_2)$ . Suppose  $S(\beta_1) = u_1$  and  $S(\beta_2) = u_2$ . By inductive hypothesis,  $N_1 = \{n \mid s_n(\beta_1) = u_1\} \in D$  and  $N_2 = \{n \mid s_n(\beta_2) = u_2\} \in D$ . Since  $D$  is a filter, the intersection  $N_1 \cap N_2 \in D$ . Hence, the set  $K = \{k \mid s_k(\beta_1) = u_1 \text{ and } s_k(\beta_2) = u_2\} \in D$ . We show that  $M = \{m \mid s_m(\beta_1 \wedge \beta_2) = u_1 \wedge u_2\} \in D$ . We prove this by showing that  $M = K$ . Let  $m \in M$ . Thus,  $s_m(\beta_1 \wedge \beta_2) = u_1 \wedge u_2$ . By definition of  $\wedge$ ,  $s_m(\beta_1 \wedge \beta_2) = s_m(\beta_1) \wedge s_m(\beta_2)$ , and therefore  $s_m(\beta_1) = u_1$  and  $s_m(\beta_2) = u_2$ . Then,  $s_m \in N_1$  and  $s_m \in N_2$ , and hence  $s_m \in K$ . Assume that  $k \in K$ . By definition of  $K$ ,  $s_k(\beta_1) = u_1$  and  $s_k(\beta_2) = u_2$ . By definition of  $\wedge$ ,  $s_k(\beta_1 \wedge \beta_2) = u_1 \wedge u_2$ . Hence,  $k \in M$ .

$(\vee)$  The proof is analogous to the previous case.

("If") The base case also follows directly from Definition 8. Assume that the theorem holds for any formula  $\beta$  with at most  $k$  connectives. *Inductive step:* The proof is done by cases on the main connective of  $\alpha$ .

$(\neg)$   $\alpha$  is of the form  $\neg\beta$  for some  $\beta$ . Let  $u$  be such that the set  $N = \{n \mid s_n(\neg\beta) = u\} \in D$ . By definition of  $\neg$ , the set  $\{n \mid \neg s_n(\beta) = u\} \in D$ . We show that the set  $M =$

$\{m \mid s_m(\beta) = \neg u\} \in D$ . For suppose that  $M \notin D$ . By the property of the ultrafilter, the complement set  $\{k \mid s_k(\beta) \neq \neg u\} \in D$ . By definition of *neg*, the set  $\{k \mid \neg s_k(\beta) \neq \neg u\} \in D^3$ . Therefore the set  $\{k \mid s_k(\neg\beta) \neq u\} \in D$ , which is a contradiction since its complement belongs to  $D$ .

( $\wedge$ )  $\alpha$  is of the form  $\beta_1 \wedge \beta_2$  for some  $\beta_1$  and  $\beta_2$ . Let  $u$  be such that the set  $N = \{n \mid s_n(\beta_1 \wedge \beta_2) = u\} \in D$ . Let  $u_1$  and  $u_2$  be such that  $M_1 = \{n \mid s_n(\beta_1) = u_1\} \in D$  and  $M_2 = \{n \mid s_n(\beta_2) = u_2\} \in D$ . By the inductive hypothesis  $S(\beta_1) = u_1$  and the  $S(\beta_2) = u_2$ , which implies that  $S(\alpha) = u_1 \wedge u_2$ . Furthermore, since  $D$  is a filter, the set  $\{n \mid s_n(\beta_1) = u_1 \text{ and } s_n(\beta_2) = u_2\} \in D$ . Therefore, the set  $\{n \mid s_n(\beta_1 \wedge \beta_2) = u_1 \wedge u_2\} \in D$ . Suppose that  $u_1 \wedge u_2 \neq u$ . Then by the property of the ultrafilter the complement  $\{n \mid s_n(\beta_1 \wedge \beta_2) \neq u_1 \wedge u_2\} \notin D$ , and hence  $N \notin D$ , which is a contradiction. Therefore,  $S(\alpha) = u_1 \wedge u_2 = u$ .

( $\vee$ ) The proof is analogous to the previous case.  $\square$

**Proposition 2** *Let  $s$  be a set-up and  $\Delta$  an infinite set of formulae. There exists  $\Delta_n \subset \Delta$ ,  $\Delta_n$  finite, such that for all  $\Delta_m \supseteq \Delta_n$ ,  $s(\Delta_m) = s(\Delta_n)$ .*

**Proof** Suppose that for all  $\Delta_n$ , there exists  $\Delta_m \supseteq \Delta_n$  such that  $s(\Delta_m) \neq s(\Delta_n)$ . Since  $s(\Delta_m) \leq s_m(\Delta_n)$ , it follows that  $s(\Delta_m) < s(\Delta_n)$ . This will lead to a infinite (strict) descending sequence of values in  $\mathbf{4}$ , which is a contradiction, since  $\mathbf{4}$  is finite.  $\square$

**Theorem 2 (Finiteness)** *Let  $\Delta$  be an infinite Belnap theory and  $\varphi$  a Belnap formula. If  $\Delta \rightarrow \varphi$ , then there exists  $\Delta' \subset \Delta$  s.t.  $\Delta'$  is finite and  $\Delta' \rightarrow \varphi$ .*

**Proof** By contrapositive. We assume that:

$$\text{for all finite } \Delta' \subset \Delta, \Delta' \not\rightarrow \varphi. \quad (5)$$

and show that  $\Delta \not\rightarrow \varphi$ . By the definition of  $\rightarrow$  and (5) we get that  $\forall \Delta_i \subset \Delta$  there exists  $s_i$ , such that  $s_i(\Delta_i) \not\leq s_i(\varphi)$ . Let  $\delta_0, \delta_1, \dots$  be an enumeration of all formulae in  $\Delta$  and  $I$  an index set constructed as follows:  $I = \{\Delta_0, \Delta_1, \dots, \Delta_n, \dots\}$ , where  $\Delta_i = \{\delta_0, \dots, \delta_i\}$ . Let  $D$  be an ultrafilter over  $I$ , defined as  $\{\hat{\Delta}_0, \hat{\Delta}_1, \dots\}$ , where  $\hat{\Delta}_j = \{\Delta_i \mid i \geq j\}$ . Consider the ultraproduct  $S = \Pi_D s_i$ . By Theorem 1,  $S$  behaves exactly as a normal set-up, and hence, for any formula  $\alpha$   $S(\alpha) = u_\alpha$  iff  $\{m \mid s_m(\alpha) = u_\alpha\} \in D$ . We show that

$$\text{for all } \Delta' \subset \Delta, \Delta' \text{ finite, } S(\Delta') \not\leq S(\varphi). \quad (6)$$

Suppose  $\exists \Delta_n \subset \Delta$  s.t.  $S(\Delta_n) \leq S(\varphi)$ . Let  $S(\Delta_n) = u_n$  for some  $u_n$  and  $S(\varphi) = f$ , for some  $f$ . By the definition of  $S$ ,  $N = \{j \mid s_j(\Delta_n) = u_n\} \in D$  and  $F = \{k \mid s_k(\varphi) = f\} \in D$ . By construction of  $D$ , the set  $\hat{\Delta}_n = \{i \mid i \geq n\} \in D$ , and therefore the set  $M = N \cap \hat{\Delta}_n = \{m \mid m \geq n \text{ and } s_m(\Delta_n) = u_n\} \in D$ . Notice that  $\Delta_m \supset \Delta_n$  for all  $m \in M$  and therefore  $s_m(\Delta_m) \leq s_m(\Delta_n)$ . Consider the set  $F \cap M$ . Since  $D$  is a filter,  $F \cap M = \{h \mid s_h(\varphi) = f \text{ and } s_h(\Delta_n) = u_n\} \in D$ . In particular, by the construction of the ultraproduct

$$1. s_h(\Delta_h) \leq s_h(\Delta_n) = u_n \leq f$$

<sup>3</sup>Note that for any  $x \in \mathbf{4}$ ,  $\neg\neg x = x$ .

$$2. s_h(\Delta_h) \not\leq s_h(\varphi) = f$$

which is a contradiction. By Proposition 2, there exists  $\Delta_n \in I$ , such that for all  $m \geq n$ ,  $S(\Delta_m) = x$  for a certain  $x \in \mathbf{4}$ . Therefore  $S(\Delta) = x$  and by (6),  $x \not\leq f$ . Hence we have constructed a set-up  $S$  such that  $S(\Delta) \not\leq S(\varphi)$  which contradicts the initial hypothesis.  $\square$

## 6 Translating Belnap's four-valued logic into classical logic

In this section, we describe a translation approach of Belnap logic into first-order logic and show that it is sound and complete with respect to Belnap's semantic notion of entailment. Let  $\mathcal{L}$  be a two sorted first-order language composed of the sort  $\mathcal{F}$ , called *B-formulae*, and the sort  $\mathcal{V}$  called *truth values*.

The set of constants of the sort  $\mathcal{F}$  is the set of propositional letters in Belnap's logic, whereas terms of  $\mathcal{F}$  are constructed using three main functions  $\neg$ ,  $\wedge$ , and  $\vee$  which correspond to the connectives in  $\mathcal{L}_B$ . The set of ground terms of  $\mathcal{F}$  is therefore equivalent to the set of Belnap wffs. The sort  $\mathcal{V}$  is instead composed of two constant symbols  $\{\text{tt}, \text{ff}\}$ , the basic constants from which Belnap's four-valued semantics can be constructed.  $\mathcal{L}$  also contains the two-sorted binary predicate *holds*. *holds* takes as first arguments,  $\mathcal{F}$  terms, and as second arguments  $\mathcal{V}$  terms.  $\mathcal{F}$  variables will be denoted with  $x, y, z, \dots$ . First-order formulae are constructed in the usual way.

Ground atomic formulae can be of two types *holds*( $\varphi, \text{tt}$ ) and *holds*( $\psi, \text{ff}$ ) for any Belnap wffs  $\varphi$  and  $\psi$ . Atomic formulae of the first type mean that " $\text{tt} \in v(\varphi)$ ", for some  $\mathbf{4}$ -valuation  $v$ , which is equivalent to say that for some  $\mathbf{4}$ -valuation,  $\varphi$  is at least true. Atomic formulae of the second type state instead that " $\text{ff} \in v(\psi)$ ", for some  $\mathbf{4}$ -valuation  $v$ , which is equivalent to say that for some  $\mathbf{4}$ -valuation,  $\psi$  is at least false. With these two types of atomic formulae it is possible to express Belnap's full four-valued semantics. For example, to say that a formula  $\alpha$  has value *Both* under some  $\mathbf{4}$ -valuation  $v$ , we can write *holds*( $\alpha, \text{tt}$ )  $\wedge$  *holds*( $\alpha, \text{ff}$ ).

The semantic behavior of Belnap connectives is fully captured by the following first-order axiomatisation.

**Definition 9** Given the two languages  $\mathcal{L}_B$  and  $\mathcal{L}$ ,  $\mathcal{A}_B$  is the first-order axiomatisation of Belnap four-valued semantics given by the following six axioms:

$$\forall x[\text{holds}(x, \text{ff}) \leftrightarrow \text{holds}(\neg x, \text{tt})] \quad (\text{Ax } 1)$$

$$\forall x[\text{holds}(x, \text{tt}) \leftrightarrow \text{holds}(\neg x, \text{ff})] \quad (\text{Ax } 2)$$

$$\forall x, y[\text{holds}(x \wedge y, \text{tt}) \leftrightarrow (\text{holds}(x, \text{tt}) \wedge \text{holds}(y, \text{tt}))] \quad (\text{Ax } 3)$$

$$\forall x, y[\text{holds}(x \wedge y, \text{ff}) \leftrightarrow (\text{holds}(x, \text{ff}) \vee \text{holds}(y, \text{ff}))] \quad (\text{Ax } 4)$$

$$\forall x, y[\text{holds}(x \vee y, \text{tt}) \leftrightarrow (\text{holds}(x, \text{tt}) \vee \text{holds}(y, \text{tt}))] \quad (\text{Ax } 5)$$

$$\forall x, y[\text{holds}(x \vee y, \text{ff}) \leftrightarrow (\text{holds}(x, \text{ff}) \wedge \text{holds}(y, \text{ff}))] \quad (\text{Ax } 6)$$

The *translation function*  $\tau$  is a mapping from the set of Belnap wffs to the set of ground atomic first-order formulae of the form *holds*( $\varphi, \text{tt}$ ). For a given Belnap formula  $\varphi$ , its first-order translation, denoted with  $\tau(\varphi)$  or simply  $\varphi^\tau$ , is the first-order atomic formula *holds*( $\varphi, \text{tt}$ ). Belnap theories are translated into a classical theory whose (first-order atomic) formulae are the translation of the Belnap formulae included in the theory. In the specific case of finite Belnap theories (i.e.

finite sets of Belnap formulae), the translation can equivalently be given by the translation of the conjunction of all Belnap formulae included in the theory. For instance, let  $\Gamma = \{\varphi_1, \dots, \varphi_n\}$  be a finite Belnap theory, its translation  $\tau(\Gamma)$ , or  $\Gamma^\tau$ , can be given by the atomic first-order formula  $holds(\varphi_1 \wedge \dots \wedge \varphi_n, \mathbf{tt})$ . This is the approach used in Theorem 3.

We are now going to prove that the above translation function together with the axiomatisation  $\mathcal{A}_B$  is sound and complete with respect to Belnap semantic notion of entailment.

**Theorem 3 (Correspondence)** *Let  $\psi$  and  $\varphi$  be two Belnap formulae. Then  $\psi \rightarrow \varphi$  if and only if  $\mathcal{A}_B, holds(\psi, \mathbf{tt}) \vdash holds(\varphi, \mathbf{tt})$  and  $\mathcal{A}_B, \neg holds(\psi, \mathbf{ff}) \vdash \neg holds(\varphi, \mathbf{ff})$ .*

The above statement captures, in first-order terms, the notion of entailment given in Definition 5 whenever  $\psi$  is of the form  $\alpha_1 \wedge \dots \wedge \alpha_n$ , with  $\{\alpha_1, \dots, \alpha_n\}$  being a Belnap theory. More specifically, for the first conjunct of the statement, the assumption  $holds(\psi, \mathbf{tt})$  is equivalent, by axiom (Ax 3), to  $holds(\alpha_1, \mathbf{tt}) \wedge \dots \wedge holds(\alpha_n, \mathbf{tt})$ , which can be read as “all  $\alpha_i$ , for each  $1 \leq i \leq n$ , are at least true”. The consequence  $holds(\varphi, \mathbf{tt})$  can also be read as  $\varphi$  is at least true. Analogously, for the second conjunct in the statement, the assumption  $\neg holds(\psi, \mathbf{ff})$  is equivalent, by axiom (Ax 4), to  $\neg holds(\alpha_1, \mathbf{ff}) \wedge \dots \wedge \neg holds(\alpha_n, \mathbf{ff})$ , where each  $\neg holds(\alpha_i, \mathbf{ff})$  can be read as “ $\alpha_i$  is not false”.

The proof of Theorem 3 uses Lemmas 1 and 2, whose proofs can be found in [Gabbay *et al.*, 2000].

**Lemma 1 (Completeness)** *Let  $\psi$  and  $\varphi$  be two Belnap formulae. If  $\psi \rightarrow \varphi$ , then  $\mathcal{A}_B, holds(\psi, \mathbf{tt}) \vdash holds(\varphi, \mathbf{tt})$  and  $\mathcal{A}_B, \neg holds(\psi, \mathbf{ff}) \vdash \neg holds(\varphi, \mathbf{ff})$ .*

**Lemma 2 (Soundness)** *Let  $\psi$  and  $\varphi$  be two Belnap formulae. If  $\mathcal{A}_B, holds(\psi, \mathbf{tt}) \vdash holds(\varphi, \mathbf{tt})$  and  $\mathcal{A}_B, \neg holds(\psi, \mathbf{ff}) \vdash \neg holds(\varphi, \mathbf{ff})$ , then  $\psi \rightarrow \varphi$ .*

**Proof of Theorem 3.** The “if-part” is given by Lemma 2 whereas the “only-if” part is given by Lemma 1.  $\square$

## 7 Revising Belnap’s four-valued logic

In this section we illustrate our belief revision approach for non-classical logics by considering in detail Belnap’s four-valued logic. Part of our motivation here is to analyse the role of inconsistency in the revision process. We have already mentioned that the result of the revision process is always consistent provided that the revising formula is not itself contradictory (K\*5). More importantly, if the old theory is consistent with the new formula, then no *true* revision is performed, i.e., the revision is equivalent to an expansion. This may cause problems if the notion of consistency of the object language differs significantly to that of classical logic. In such cases, we need to seek for alternatives in order to *trigger* the revision process in the translation.

Our view is that one does not need to be limited to the notion of consistency in the object language, as long as some notion of *acceptability* of theories in the object language is defined via classical logic axioms.

The idea now is to map the notion of acceptability seen in Definition 3 to classical logic in such a way that the translation of *unacceptable* Belnap theories yields *inconsistent* classical logic theories, when then a true revision can be performed. We also need to ensure that the acceptability notion is included in the revising part, so that by (K\*2), the result of the revision will be itself *acceptable*, as long as the input is not itself unacceptable. Our notion of acceptability can be translated to classical logic in the following way:

**Definition 10** Acceptability axiom for Belnap theories:

$$Acc = \forall \alpha [(holds(\alpha, \mathbf{tt}) \wedge holds(\alpha, \mathbf{ff})) \rightarrow \perp] \quad \square$$

As we expected, the co-existence of contradictory (Belnap) information will not be tolerated by the revision process. This will force propositions to have *at most* one of the truth-values  $\mathbf{tt}$  and  $\mathbf{ff}$ . Intuitively, this filters out Belnap consequences with truth-value Both.

In order to simplify notation, the following convention will be used:

$$\begin{aligned} holds(\alpha, \mathbf{tt}) &= \alpha^+ & holds(\neg\alpha, \mathbf{tt}) &= \overline{\alpha}^+ \\ holds(\alpha, \mathbf{ff}) &= \alpha^- & holds(\neg\alpha, \mathbf{ff}) &= \overline{\alpha}^- \end{aligned}$$

Furthermore, we use  $\neg\alpha^+$  for  $\neg holds(\alpha, \mathbf{tt})$ , etc.

Belief revision in Belnap’s logic is defined as follows:

**Definition 11** [Belief Revision in Belnap’s four-valued logic] Let  $\psi$  and  $\varphi$  be two Belnap formulae and  $\psi^+$  ( $\psi^-$ ) and  $\varphi^+$  ( $\varphi^-$ ) their translation to classical logic as described above, respectively. Let  $\mathcal{A}_B$  be as in Definition 9 and  $Acc$  as in Definition 10. The revision of  $\psi$  by  $\varphi$ , in symbols  $\psi *_b \varphi$ , is defined as

$$\begin{aligned} \psi *_b \varphi &= \{ \gamma \mid \psi^+ *_a (\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \gamma^+ \\ &\text{and } \neg \psi^- *_a (\neg \varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg \gamma^- \} \quad \square \end{aligned}$$

The compound statement in the definition is necessary because of the way the translation was done. In fact, the positive  $holds$  checks the relevant part of one half of the lattice **L4**, i.e., Both and T, whereas the negative  $holds$  checks the other half, i.e., None and T. F needs not to be checked since it is less than or equal any value in **L4**.

Notice that, without  $Acc$ , the revision of  $\psi$  by  $\varphi$  is simply  $\psi \wedge \varphi$ . The correspondence axiom tell us that  $\psi \wedge \varphi \rightarrow \gamma$  iff  $\mathcal{A}_B \wedge (\psi \wedge \varphi)^+ \vdash \gamma^+$  and  $\mathcal{A}_B \wedge \neg(\psi \wedge \varphi)^- \vdash \neg \gamma^-$ . Since  $\varphi^+ \wedge \mathcal{A}_B$  is consistent with  $\psi^+$ ,  $\psi^+ *_a (\varphi^+ \wedge \mathcal{A}_B)$  is just  $\psi^+ \wedge \varphi^+ \wedge \mathcal{A}_B$  (the same for the negative part). The revision then gives us all  $\gamma$ ’s such that  $\psi^+ \wedge \varphi^+ \wedge \mathcal{A}_B \vdash \gamma^+$  and  $\neg \psi^- \wedge \neg \varphi^- \wedge \mathcal{A}_B \vdash \neg \gamma^-$ . Notice that in the presence of  $\mathcal{A}_B$ ,  $\psi^+ \wedge \varphi^+ \leftrightarrow (\psi \wedge \varphi)^+$  and  $\neg \psi^- \wedge \neg \varphi^- \leftrightarrow \neg(\psi \wedge \varphi)^-$  (from (Ax 3) and (Ax 4), respectively). Therefore, we get all  $\gamma$ ’s such that  $\mathcal{A}_B \wedge (\psi \wedge \varphi)^+ \vdash \gamma^+$  and  $\mathcal{A}_B \wedge \neg(\psi \wedge \varphi)^- \vdash \neg \gamma^-$ , which is  $\{ \gamma \mid \psi \wedge \varphi \rightarrow \gamma \}$ .

### 7.1 AGM postulates under Belnap’s perspective

(K\*1)  $\psi *_b \varphi$  is a belief set

In our terms, this means that the set of Belnap formulae that we receive back from the revision process should be closed under  $\rightarrow$ . In other words,  $\gamma \in \psi *_b \varphi$  iff  $\psi *_b \varphi \rightarrow \gamma$ . (only if) if  $\gamma \in \psi *_b \varphi$ , it follows by reflexivity of  $\rightarrow$  that  $\psi *_b \varphi \rightarrow \gamma$ . (if) Suppose that  $\psi *_b \varphi \rightarrow \gamma$ , we need to show that  $\gamma \in \psi *_b \varphi$ . By the correspondence theorem, if  $\psi *_b \varphi \rightarrow \gamma$ , then

$\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \vdash \gamma^+$  and  $\{\mathcal{A}_B\} \cup \{\neg\alpha^- \mid \neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\alpha^-\} \vdash \neg\gamma^-$

Notice that  $\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \subseteq \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  since by (K\*2),  $\mathcal{A}_B \in \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  and by (K\*1),  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  is closed under classical logical consequence and hence contains all such  $\alpha^+$ . By monotonicity of classical logic, it follows that  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \gamma^+$ . By a similar argument, we can show that  $\neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\gamma^-$  and hence  $\gamma \in \psi^*_b\varphi$ .

(K\*2)  $\varphi \in \psi^*_b\varphi$

Remember that by (K\*2),  $\varphi^+ \wedge \mathcal{A}_B \wedge Acc \in \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  and  $\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc \in \neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc)$ . By (K\*1),  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \varphi^+ \wedge \mathcal{A}_B \wedge Acc$  and hence  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \varphi^+$ . Similarly,  $\neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\varphi^-$ . By Definition 11,  $\varphi \in \psi^*_b\varphi$ .

(K\*3) and (K\*4): we argued in Section 2, page 2, that for non-classical logics, the two postulates above have the meaning expressed by (K\*<sub>3,4</sub>) below:

(K\*<sub>3,4</sub>) If  $A$  is consistent with  $K$ , then  $K *_a A = \text{Cn}_b(K \cup \{A\})$ .

In this case, being “consistent” means *acceptable*. In other words, that acceptability plays no role in the revision process. The equivalence above was discussed previously, just before the beginning of Section 7.1.

(K\*5)  $\psi^*_b\varphi = \psi_\perp$  only if  $\varphi$  is contradictory

Similarly, this should have the following interpretation “ $\psi^*_b\varphi$  is unacceptable, only if  $\varphi$  is (itself) unacceptable.”

This can be shown, if we remember that we artificially forced the translation of unacceptable Belnap theories to be classically inconsistent. This means, for instance, that if  $\varphi$  is not itself unacceptable, then  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  is classically consistent.

Suppose  $\psi^*_b\varphi$  is unacceptable, we must show that  $\varphi$  is unacceptable. From the assumption, it follows that

$$\psi^*_b\varphi \rightarrow \gamma \text{ and } \psi^*_b\varphi \rightarrow \neg\gamma \quad (7)$$

for some  $\gamma$  in  $\mathcal{L}_B$ . By the correspondence theorem and (7),  $\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \vdash \gamma^+$ .

Notice that  $\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \subseteq \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  since by (K\*2),  $\mathcal{A}_B \in \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  and by (K\*1),  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  is closed under classical logical consequence and hence contains all such  $\alpha^+$ . Also notice that by the correspondence theorem and (7),  $\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \vdash \bar{\gamma}^+$ . For the same reason,  $\{\mathcal{A}_B\} \cup \{\alpha^+ \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+\} \subseteq \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  and therefore  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \gamma^+ \wedge \bar{\gamma}^+ = \text{holds}(\gamma, \text{tt}) \wedge \text{holds}(\neg\gamma, \text{tt})$ .

Now  $\mathcal{A}_B \wedge \neg\text{holds}(\neg\gamma, \text{tt}) \vdash \text{holds}(\gamma, \text{ff})$ . By (K\*1),  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \text{holds}(\gamma, \text{tt}) \wedge \text{holds}(\gamma, \text{ff})$ .  $Acc \in \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \therefore \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \perp$ . By (K\*5),  $(\varphi^+ \wedge \mathcal{A}_B \wedge Acc)$  must be contradictory.  $\mathcal{A}_B$  and  $Acc$  do not introduce themselves any inconsistencies, so it must be the case that  $\varphi$  is itself unacceptable.

(K\*6) If  $\varphi \equiv_b \gamma$ , then  $\psi^*_b\varphi \equiv_b \psi^*_b\gamma$

In Belnap’s logic,  $\varphi \equiv_b \gamma$  means  $\varphi \rightarrow \gamma$  and  $\gamma \rightarrow \varphi$ . By the correspondence:  $\mathcal{A}_B \vdash \text{holds}(\varphi, \text{tt}) \leftrightarrow \text{holds}(\gamma, \text{tt})$

and also  $\mathcal{A}_B \vdash \neg\text{holds}(\varphi, \text{ff}) \leftrightarrow \neg\text{holds}(\gamma, \text{ff})$ . We show the positive part only. The negative part is similar. It is easy to see that  $\vdash (\mathcal{A}_B \wedge \varphi^+) \leftrightarrow (\mathcal{A}_B \wedge \gamma^+)$ . By (K\*6),  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \equiv \psi^{+*}_a(\gamma^+ \wedge \mathcal{A}_B \wedge Acc)$ . The same happens with the negative part and therefore, by the definition of revision in Belnap’s logic,  $\psi^*_b\varphi = \psi^*_b\gamma$ , and hence  $\psi^*_b\varphi \equiv_b \psi^*_b\gamma$ .

(K\*7) and (K\*8): we argued in Section 2, page 2, that these two postulates have the meaning expressed by (K\*<sub>7,8</sub>) below:

(K\*<sub>7,8</sub>)  $\text{Cn}_b(\psi^*_b\varphi \cup \{\gamma\}) = \psi^*_b(\varphi \wedge \gamma)$ , when  $\gamma$  is consistent with  $\psi^*_b\varphi$

In this case, being “consistent” means *acceptable*. Acceptability plays therefore no role in the revision process above, and hence  $\psi^*_b\varphi \cup \{\gamma\}$  is *acceptable*. In first order logic terms, this means that the acceptability axiom is never used in the derivation of translated Belnap formulae.

We sketch the proof as follows. Let us look first at the formulae in the set  $\psi^*_b\varphi \cup \{\gamma\}$ . These are the formulae in the set  $\{\alpha \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+ \text{ and } \neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\alpha^-\} \cup \{\gamma\}$ . On the other hand, the formulae in the set  $\psi^*_b(\varphi \wedge \gamma)$  are the same formulae in the set  $\{\beta \mid \psi^{+*}_a((\varphi \wedge \gamma)^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \beta^+ \text{ and } \neg\psi^{-*}_a(\neg(\varphi \wedge \gamma)^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\beta^-\}$ . Now by the assumption and (K\*<sub>7,8</sub>) for  $*_a$ , we have that  $\text{Cn}(\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \cup \{\gamma^+\}) = \psi^{+*}_a((\varphi \wedge \gamma)^+ \wedge \mathcal{A}_B \wedge Acc)$ . Notice that  $\mathcal{A}_B \in \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \cup \{\gamma^+\}$ . It is easy to show that  $\psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \cup \{\gamma^+\} \vdash \beta^+$  and  $\neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \cup \{\neg\gamma^-\} \vdash \neg\beta^-$  iff  $\beta \in \text{Cn}_b(\{\alpha \mid \psi^{+*}_a(\varphi^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+ \text{ and } \neg\psi^{-*}_a(\neg\varphi^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\alpha^-\} \cup \{\gamma\})$ . Similarly,  $\psi^{+*}_a((\varphi \wedge \gamma)^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \beta^+$  and  $\neg\psi^{-*}_a(\neg(\varphi \wedge \gamma)^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\beta^-$  iff  $\beta \in \text{Cn}_b(\{\alpha \mid \psi^{+*}_a((\varphi \wedge \gamma)^+ \wedge \mathcal{A}_B \wedge Acc) \vdash \alpha^+ \text{ and } \neg\psi^{-*}_a(\neg(\varphi \wedge \gamma)^- \wedge \mathcal{A}_B \wedge Acc) \vdash \neg\alpha^-\})$ .

## 8 Conclusions

We have presented a way of exporting an AGM revision process in classical logic to non-classical logics. The approach can be used for any non-classical logic whose semantics is first-order logic axiomatisable. Moreover, as shown in Proposition 2, the approach here described is also applicable to non-classical logics whose underlying semantic lattice does not have an infinite decreasing chain of truth-values. Further work is still needed in order to apply the approach to other non-classical logics, such as Fuzzy logic, where the underlying structure does not respect this property.

There are considerable benefits for adopting our approach. Various translation mechanisms have been developed for many non-classical logics, which allow to mimic the non-classical logical reasoning within the context of classical logic. Existing results in the literature have already shown the benefits that such translations mechanisms can provide, e.g., to give an object logic a semantics [Broda *et al.*, 1999], to compare it with other logics, to get decidability/undecidability results, to make use of automated deduction of classical logic, etc. Our approach can equally benefit from building upon a translation method and classical AGM belief revision, as it can allow to make use of existing automated systems for classical belief revision but in the context

of non-classical logics. Moreover, revision theory for classical logic is very well developed. There are various notions and fine tuning involved, and using translation mechanisms into classical logic will not only open a wealth of distinctions for the source logic but also enrich classical logic revision itself with new ideas and problems arising from non-classical logics.

Furthermore, from the point of view of classical logic, our approach is a first example of a more general revision process, called *relative revision*. This concept can be defined as follows. Given a theory  $T$  of classical logic (for example a theory of linearly ordered Abelian groups, we can talk about a set of sentences  $\Delta$  being acceptable (i.e.,  $\text{Cn}(T \cup \Delta)$  being acceptable)<sup>4</sup>. We can talk about revision relative to  $T$  of  $\Delta$  by  $\psi$  (denoted by  $\Delta *_T \psi$ ) yielding a new acceptable theory  $\Delta' = \Delta *_T \psi$ . The case where  $T$  arises in connection with a translation from another logic is only one instance of this general relative revision. A complete and detail framework for relative revision is subject of our future work.

## References

- [Belnap, 1977a] N. Belnap. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–56. Oriel Press, 1977.
- [Belnap, 1977b] N. Belnap. A useful four-valued logic. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-valued Logic*, pages 8–37. D. Reidel, 1977.
- [Broda *et al.*, 1999] Krysia Broda, Alessandra Russo, and Dov Gabbay. *Discovering World with Fuzzy Logic: Perspectives and Approaches to the Formalisation of Human-Consistent Logical Systems*, chapter A Unified Compilation Style Labelled Deductive Systems for Modal, Substructural and Fuzzy Logic. Springer Verlag, 1999.
- [Darwiche and Pearl, 1996] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. Technical Report R-202, Cognitive Science Laboratory, Computer Science Department, University of California, Los Angeles, CA 90024, November 1996.
- [Darwiche and Pearl, 1997] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–29, 1997.
- [Gabbay *et al.*, 2000] Dov Gabbay, Odinaldo Rodrigues, and Alessandra Russo. *Information, Uncertainty, Fusion*, chapter Revision by Translation, pages 3–31. Kluwer Academic Publishers, 2000. ISBN: 0-7923-8590-X.
- [Gärdenfors, 1988] Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. A Bradford Book - The MIT Press, Cambridge, Massachusetts - London, England, 1988.
- [Ohlbach, 1991] H.J. Ohlbach. Semantics-based translations methods for modal logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.
- [Restall and Slaney, 1995] Greg Restall and John Slaney. Realistic belief revision. Technical Report TR-ARP-2-95, Australian National University, Australian National University, January 1995.

---

<sup>4</sup>Let  $\mathcal{M}$  be a class of models of  $T$ .  $\Delta$  is  $T$ -consistent if  $\Delta \cup T$  has a model.  $\Delta$  is  $\mathcal{M}$ -acceptable if  $\Delta \cup T$  has a model in  $\mathcal{M}$ . Given  $\Delta$  and  $\psi$ , we want the result of the revision of  $\Delta$  by  $\psi$  to be  $\mathcal{M}$ -acceptable.