Belief Revision: An Introduction

PETER GÄRDENFORS

Cognitive Science, Department of Philosophy,
Lund University, S-223 50 Lund, Sweden

1 THE PROBLEMS OF BELIEF REVISION

1.1 An Example

Suppose that you have a database that contains, among other things, the following pieces of information (in some form of code):

α: All European swans are white.
β: The bird caught in the trap is a swan.
γ: The bird caught in the trap comes from Sweden.
δ: Sweden is part of Europe.

If your database is coupled with a program that can compute logical inferences in the given code, the following fact is derivable from α - δ:

ε: The bird caught in the trap is white.

Now suppose that, *as a matter of fact*, the bird caught in the trap turns out to be black. This means that you want to add the fact ¬ε, i.e., the negation of ε, to the database. But then the database becomes *inconsistent*. If you want to keep the database consistent, which is normally a sound methodology, you need to *revise* it. This means that some of the beliefs in the original database must be retracted. You don't want to give up all of the beliefs since this would be an unnecessary loss of valuable information. So you have to *choose* between retracting α, β, γ or δ.

The problem of belief revision is that logical considerations alone do not tell you which beliefs to give up, but this has to be decided by some other means. What makes things more complicated is that beliefs in a database have *logical consequences*, so when giving up a belief you have to decide as well which of the consequences to retain and which to retract. For example, if you decide to retract α in the situation described here, α has as logical consequences, among others, the following two:

α': All European swans except the one caught in the trap are white

and

α": All European swans except some of the Swedish are white.

Do you want to keep any of these sentences in the revised database?

1.2 The Methodological Problems of Belief Revisions

When trying to handle belief revisions in a computational setting, there are three main methodological questions to settle:

(1) How are the beliefs in the database *represented*?

Most databases work with elements like *facts* and *rules* as primitive forms of representing information. The code used to represent the beliefs may be more or less closely related to standard logical formalism. A mechanism for belief revision is sensitive to the formalism chosen to represent the beliefs.

(2) What is the relation between the elements explicitly represented in the database and the beliefs that may be *derived* from these elements?

This relation is to a large extent dependent on the *application area* of the database. In some cases the elements explicitly formulated in the database have a special status in comparison to the logical consequences of these beliefs that may be derived by some inference mechanism. In other cases, the formulation of the beliefs in the database is immaterial so that any representation that has the same logical consequences, i.e., the same set of implicit beliefs, is equivalent. As will be seen in several papers in this volume, the nature of the relation between explicit and implicit beliefs is of crucial importance for how the belief revision process is attacked.

(3) How are the choices concerning what to retract made?

Logic alone is not sufficient to decide between which beliefs to give up and which to retain when performing a belief revision. What are the extralogical factors that determine the choices? One idea is that the information lost when giving up beliefs should be kept minimal. Another idea is that some beliefs are considered more important or entrenched than others and the beliefs that should be retracted are the least important ones. Within computer science the use of *integrity constraints* is a common way of handling the problem. Again, the methodological rules chosen here are dependent on the application area.

1.3 Three Kinds of Belief Changes

A belief revision occurs when a new piece of information that is *inconsistent* with the present belief system (or database) is added to that system in such a way that the result is a new consistent belief system. But this is not the only kind of change that can occur in a belief system. Depending on how beliefs are represented and what kinds of inputs are accepted, different typologies of belief changes are possible.

In the most common case, when beliefs are represented by *sentences* in some code, and when a belief is either *accepted* or *rejected* in a belief system K (so that no degrees of belief are considered), one can distinguish three main kinds of belief changes:

(i) *Expansion*: A new sentence ϕ is added to a belief system K together with the logical consequences of the addition (regardless of whether the larger set so formed is consistent). The belief system that results from expanding K by a sentence ϕ will be denoted $K+\phi$.

(ii) *Revision*: A new sentence that is inconsistent with a belief system K is added, but, in order to maintain consistency in the resulting belief system, some of the old sentences in K are deleted. The result of revising K by a sentence ϕ will be denoted $K\dotplus\phi$.

(iii) *Contraction*: Some sentence in K is retracted without adding any new facts. In order for the resulting system to be closed under logical consequences some other sentences from K must be given up. The result of contracting K with respect to ϕ will be denoted $K\dot{-}\phi$.

Expansions of belief systems can be handled comparatively easily. $K+\phi$ can simply be defined as the logical closure of K together with ϕ:

(Def +) $\qquad K+\phi = \{\psi: K \cup \{\phi\} \vdash \psi\}$

As is easily shown, $K+\phi$ defined in this way will be closed under logical consequences and will be consistent when ϕ is consistent with K.

It is not possible to give a similar explicit definition of revisions and contractions in logical and set-theoretical notions only. The problems for revisions were presented in the introductory example. There is no purely logical reason for making one choice rather than the other among the sentences to be retracted, but we have to rely on additional information about these sentences. Thus, from a logical point of view, there are several ways of specifying the revision $K\dotplus\phi$. Though $K\dotplus\phi$ cannot be characterized uniquely in logical terms, the *general properties* of a revision function can be investigated, and – in some cases, at least – *algorithms* can be found for computing revision functions. These two goals will be handled technically by using the notion of a *revision function* "$\dotplus$" which has two arguments, a belief system K and a sentence ϕ, and which has as its value the revised belief system $K\dotplus\phi$.

The contraction process faces parallel problems. To give a simple example, consider a belief system K which contains the sentences ϕ, ψ, $\phi \wedge \psi \rightarrow \chi$ and their logical consequences (among which is χ). Suppose that we want to contract K by deleting χ. Of course, χ must be deleted from K when forming $K \dot{-} \chi$, but also at least one of the sentences ϕ, ψ, or $\phi \wedge \psi \rightarrow \chi$ must be given up in order to maintain consistency. Again, there is no purely logical reason for making one choice rather than the other. Another concrete example is provided by Fagin, Ullman and Vardi (1983, p. 353).

The common denominator in both this example and the introductory one is that the database is not viewed merely as a collection of logically independent facts, but rather as a collection of axioms from which other facts can be derived. It is the interaction between the updated facts and the derived facts that is the source of the problem.

In parallel with revision we can introduce the concept of a *contraction function* "$\dot{-}$" which has the same two arguments as before, i.e., a belief system K and a sentence ϕ (to be retracted from K), and which produces as its value the belief system $K \dot{-} \phi$. In Section 3.3, I shall show that the problems of revision and contraction are closely related – being two sides of the same coin.

1.4 Two Approaches to Describing Belief Revisions

When tackling the problem of belief revision there are two general strategies to follow, namely, to present explicit *constructions* of the revision process and to formulate *postulates* for such constructions. For a computer scientist the ultimate solution to the problem about belief revision is to develop *algorithms* for computing appropriate revision and contraction functions for an arbitrary belief system. In this volume several proposals for constructions of revision methods will be presented. These methods are not presented as pure algorithms, but on a slightly more general level.

However, in order to know whether an algorithm is successful or not it is necessary to determine what an 'appropriate' revision function is. Our standards for revision and contraction functions will be various *rationality postulates*. The formulations of these postulates are given in a more or less equational form. One guiding idea is that the revision $K \dot{+} \phi$ of K with respect to ϕ should represent the minimal change of K needed to accommodate ϕ consistently. The consequences of the postulates will also be investigated.

Much of the theoretical work within belief revision theory consists of connecting the two approaches. This is done via a number of *representation theorems*, which show that the revision methods that satisfy a particular set of rationality postulates are exactly those that fall within some computationally well defined class of methods.[1]

[1] For further discussion of the two strategies cf. Makinson (1985, pp. 350-351).

2 MODELS OF BELIEF STATES

2.1 Preliminaries

Before we can start discussing models of belief revision, we must have a way of modelling belief states since a revision method is defined as a function from one belief state into another. The most common models of belief states in computational contexts are *sentential* or *propositional*, in the sense that the elements constituting the belief systems are coded as formulas representing sentences. This kind of model will be the focus of this introduction, but some alternative types of models will be encountered in the volume.

But even if we stick to propositional models of belief systems, there are many options. First of all, we must choose an appropriate *language* to formulate the belief sentences. For example, databases include some form of *rules*, and there are many ways of formalizing these: as quantified sentences in first order logic, as PROLOG rules (corresponding to Horn-clauses), as default statements (e.g., in the style of Reiter (1980)), as probability statements, etc.

In this introduction, I shall work with a language L which is based on first order logic. The details of L will be left open for the time being. It will be assumed that L is closed under applications of the *boolean operators* $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction) and $\rightarrow$ (implication). We will use ϕ, ψ, χ, etc. as variables over sentences in L. It is also convenient to introduce the symbols $\top$ and $\bot$ for the two sentential constants "truth" and "falsity."

What is accepted in a formal model of a belief state are not only the sentences that are explicitly put into the database, but also the *logical consequences* of these beliefs. Hence, the second factor which has to be decided upon when modelling a belief state is what *logic* governs the beliefs. In practice this depends on which theorem-proving mechanism is used in combination with the database. However, when doing a theoretical analysis, one wants to abstract from the idiosyncracies of a particular algorithm for theorem proving and start from a more general description of the logic. If the logic is undecidable, further complications will arise, but we will ignore these for the time being.

I shall assume that the underlying logic includes *classical propositional logic* and that it is compact.[2] If K logically entails ϕ we will write this as $K \vdash \phi$. Where K is a set of sentences, we shall use the notation Cn(K) for the set of all logical consequences of K, i.e., $Cn(K) = \{\phi : K \vdash \phi\}$. All papers in this volume presume classical logic, except the one by Cross and Thomason where a four-valued logic is used instead.

[2] A logic is compact iff whenever A is a logical consequence of a set of sentence K, then there is a *finite* subset K' of K such that A is a logical consequence of K'.

2.2 Belief Sets

The simplest way of modelling a belief state is to represent it by a *set* of sentences from L. Accordingly, we define a *belief set* as a set K of sentences in L which satisfies the following *integrity constraint*:[3]

(I) If K logically entails ψ, then $\psi \in$ K.

In logical parlance, (I) says that K is *closed under logical consequences*. The interpretation of such a set is that it contains all the sentences that are *accepted* in the modelled belief state. Consequently, when $\phi \in$ K we say that ϕ is accepted in K and when $\neg\phi \in$ K we say that ϕ is rejected in K. It should be noted that a sentence being accepted does not imply that it has any form of justification or support.[4] A belief set can also be seen as a *theory* which is a partial description of the world. "Partial" because in general there are sentences ϕ such that neither ϕ nor $\neg\phi$ are in K.

By classical logic, whenever K is *inconsistent*, then K $\vdash \phi$ for every sentence ϕ of the language L. This means that there is exactly one inconsistent belief set under our definition, namely, the set of all sentences of L. We introduce the notation $K_\perp$ for this belief set.

2.3 Belief Bases

Against modelling belief states as belief sets it has been argued (Makinson 1985, Hansson 1990, 1991, Nebel 1990, Fuhrmann 1991) that some of our beliefs have no independent standing but arise only as inferences from our more basic belief. It is not possible to express this distinction in a belief set since there are no markers for which beliefs are basic and which are derived. Furthermore, it seems that when we perform revisions or contractions we never do it to the belief set itself which contains an infinite number of elements, but rather on some finite *base* for the belief set.

Formally, this idea can be modelled by saying that B_K is a *base for a belief set* K iff B_K is a finite subset of K and $Cn(B_K) = K$. Then instead of introducing revision and contraction functions that are defined on belief sets it is assumed that these functions are defined on bases. Such functions will be called *base revisions* and *base contractions* respectively. This approach introduces a more finegrained structure since we can have two bases B_K and C_K such that $Cn(B_K) = Cn(C_K)$ but $B_K \neq C_K$. The papers by Nebel and Hansson in this volume concern base revisions. They will be presented in Section 3.5.

[3] Belief sets were called *knowledge sets* in Gärdenfors and Makinson (1988).

[4] For further discussion of the interpretation of belief sets cf. Gärdenfors (1988).

There is no general answer to the question of which model is the best of full belief sets or bases, but this depends on the particular application area. Within computer science applications, bases seem easier to handle since they are explicitly finite structures. However, it has been argued in Gärdenfors (1990) that much of the computational advantages of bases for belief sets can be modelled by belief sets together with the notion of *epistemic entrenchment* of beliefs (cf. Section 4.1).

2.4 Possible Worlds Models

An obvious objection to using sets of sentences as models of belief states is that the *objects* of belief are normally not sentences but rather the *contents* of sentences, that is, propositions. The characterization of propositions that has been most popular among philosophers during recent years is to identify them with *sets of possible worlds*. The basic semantic idea connecting sentences with propositions is then that a sentence expresses a given proposition if and only if it is true in exactly those possible worlds that constitute the set of worlds representing the proposition.

By taking beliefs to be beliefs in propositions, we can then model a belief state by a set W_K of possible worlds. The epistemic interpretation of W_K is that it is the narrowest set of possible worlds in which the individual being in the modelled belief state is certain to find the actual world. This kind of model of a belief state has been used by Harper (1977), Grove (1988), among others and in a generalized form by Spohn (1988) (also cf. the comparisons in Gärdenfors (1978)). In this volume, Katsuno and Mendelzon, and Morreau use this way of modelling belief states.

There is a very close correspondence between belief sets and possible worlds models. For any set W_K of possible worlds we can define a corresponding belief set K as the set of those sentences that are true in all worlds in W_K (assuming that the set of propositional atoms is finite). It is easy to verify that K defined in this way satisfies the integrity constraint (I) so that it is indeed a belief set. Conversely, for any belief set K, we can define a corresponding possible worlds model W_K by identifying the possible worlds in W_K with the *maximal consistent extensions* of K. Then we say that a sentence ϕ is *true* in such an extension w iff $\phi \in w$. Again it is easy to verify that this will generate an appropriate possible worlds model (for details cf. Grove (1988)).

From a computational point of view, belief sets are much more tractable than possible worlds models. So even though possible worlds models are popular among logicians, the considerations here show that the two kinds of models are basically equivalent. And if we want to implement belief revision systems, sentential models like belief sets, and in particular bases for belief sets, are much easier to handle.

2.5 Justifications vs. Coherence Models

Another question that has to be answered when modelling a state of belief is whether the *justifications* for the beliefs should be part of the model or not. With respect to this question there are two main approaches. One is the *foundations* theory which holds that one should keep track of the justifications for one's beliefs: Propositions that have no justification should not be accepted as beliefs. The other is the *coherence* theory which holds that one need not consider the pedigree of one's beliefs. The focus is instead on the *logical* structure of the beliefs – what matters is how a belief coheres with the other beliefs that are accepted in the present state.[5] The belief sets presented above clearly fall into the latter category.

It should be obvious that the foundations and the coherence theories have very different implications for what should count as rational *changes* of belief systems. According to the foundations theory, belief revision should consist, first, in giving up all beliefs that no longer have a *satisfactory justification* and, second, in adding new beliefs that have become justified. On the other hand, according to the coherence theory, the objectives are, first, to maintain *consistency* in the revised epistemic state and, second, to make *minimal changes* of the old state that guarantee sufficient overall coherence. Thus, the two theories of belief revision are based on conflicting ideas of what constitutes rational changes of belief. The choice of underlying theory is, of course, also crucial for how a computer scientist will attack the problem of implementing a belief revision system.

Doyle's paper in this volume deals with the relations between justification theories and coherence theories of belief revision. In an earlier paper (Gärdenfors 1990), I presented some arguments for preferring the coherence approach to the foundations approach. Doyle argues that I have overemphasized the differences between the two approaches. He also wants to show that the foundations approach represents the most direct way of making the coherence approach computationally accessible.

Galliers' theory of autonomous belief revision, also in this volume, suggests in another way that the choice between coherence and foundational theories may not be exclusive; her theory in fact represents a blend between the two approaches. In a sense, also the belief base models presented in Section 2.3 show traces of justificationalism – the beliefs in the base are thought of as more foundational than the derived beliefs.

[5] Harman (1986) presents an analysis of the epistemological aspects of the two approaches.

3 RATIONALITY POSTULATES FOR BELIEF REVISION

3.1 The AGM Postulates for Revision

In this section, it will be assumed that belief sets (that is sets of sentences closed under logical consequences) are used as models of belief states. The goal is now to formulate postulates for rational revision and expansion functions defined over such belief sets.

The underlying motivation for these postulates (which are taken from Alchourrón, Gärdenfors, and Makinson (1985), hence the name) is that when we change our beliefs, we want to retain as much as possible from our old beliefs – we want to make a *minimal change*. Information is in general not gratuitous, and unnecessary losses of information are therefore to be avoided. This heuristic criterion may be called the criterion of *informational economy*.

However, it turns out to be difficult to give a precise quantitative definition of the loss of information (see, e.g., the discussion of minimality in Gärdenfors 1988, pp. 66-68). Instead we shall follow another line of specifying 'minimal change': We assume that the sentences in a belief set have different degrees of *epistemic entrenchment*, and when we give up sentences when forming a revision or a contraction, we give up those with the lowest degree of entrenchment. The idea of epistemic entrenchment will be presented in greater detail in Section 4.1.

It is assumed that for every belief set K and every sentence ϕ in L, there is a *unique* belief set $K\dotplus\phi$ representing the revision of K with respect to ϕ. In other words $\dotplus$ is a *function* taking a belief set and a sentence as arguments and giving a belief set as a result. This is admittedly a strong assumption, since in many cases, the information available is not sufficient to determine a unique revision. However, from a computational point of view this assumption is gratifying. In Doyle (1991) and Galliers' paper in this volume this assumption is not made.

The first postulate requires that the outputs of the revision function indeed be belief sets:

($K\dotplus1$) For any sentence ϕ and any belief set K, $K\dotplus\phi$ is a belief set.

The second postulate guarantees that the input sentence ϕ is accepted in $K\dotplus\phi$:

($K\dotplus2$) $\phi \in K\dotplus\phi$.

The normal application area of a revision process is when the input ϕ contradicts what is already in K, that is $\neg\phi \in$ K. However, in order to have the revision function defined for all arguments, we can easily extend it to cover the case when $\neg\phi \notin$ K. In this case, revision is identified with expansion. For technical reasons, this identification is divided into two parts:

(K∔3) $K\dotplus\phi \subseteq K+\phi$.

(K∔4) If $\neg\phi \notin K$, then $K+\phi \subseteq K\dotplus\phi$.

The purpose of a revision is to produce a new *consistent* belief set. Thus $K\dotplus\phi$ should be consistent, unless ϕ is logically impossible:

(K∔5) $K\dotplus\phi = K_{\perp}$ if and only if $\vdash \neg\phi$.

It should be the *content* of the input sentence ϕ rather than its particular linguistic formulation that determines the revision. In other words, belief revisions should be analysed on the *knowledge level* and not on the syntactic level. This means that logically equivalent sentences should lead to identical revisions:

(K∔6) If $\vdash \phi \leftrightarrow \psi$, then $K\dotplus\phi = K\dotplus\psi$.

The postulates (K∔1) - (K∔6) are elementary requirements that connect K, ϕ and $K\dotplus\phi$. This set will be called the *basic* set of postulates. The final two conditions concern *composite* belief revisions. The idea is that, if $K\dotplus\phi$ is a revision of K and $K\dotplus\phi$ is to be changed by a further sentence ψ, such a change should be made by expansions of $K\dotplus\phi$ whenever possible. More generally, the minimal change of K to include both ϕ and ψ, that is, $K\dotplus\phi\wedge\psi$, ought to be the same as the expansion of $K\dotplus\phi$ by ψ, so long as ψ does not contradict the beliefs in $K\dotplus\phi$. For technical reasons the precise formulation is split into two postulates:

(K∔7) $K\dotplus\phi\wedge\psi \subseteq (K\dotplus\phi)+\psi$.

(K∔8) If $\neg\psi \notin K\dotplus\phi$, then $(K\dotplus\phi)+\psi \subseteq K\dotplus\phi\wedge\psi$.

When $\neg\psi \in K$, then $(K\dotplus\phi)+\psi$ is $K_{\perp}$, which is why the proviso is needed in (K∔8) but not in (K∔7).

We turn next to some consequences of the postulates. It can be shown (Gärdenfors, 1988, p. 57) that in the presence of the basic set of postulates (K∔7) is equivalent to:

(1) $K\dotplus\phi \cap K\dotplus\psi \subseteq K\dotplus\phi\vee\psi$.

Another principle that is useful is the following 'factoring' condition:

(2) $K\dotplus\phi\vee\psi = K\dotplus\phi$ or $K\dotplus\phi\vee\psi = K\dotplus\psi$ or $K\dotplus\phi\vee\psi = K\dotplus\phi \cap K\dotplus\psi$.

It can be shown that, given the basic postulates, (2) is in fact equivalent to the conjunction of (K∔7) and (K∔8).

Furthermore (K∔7) and (K∔8) together entail the following identity criterion:

(3) $K\dotplus\phi = K\dotplus\psi$ if and only if $\psi \in K\dotplus\phi$ and $\phi \in K\dotplus\psi$.