

# Modelling Belief Dynamics

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.

(Peter Gärdenfors)

The following considerations concern modelling Belief Dynamics (BD) not just in the sense of a formalization, but rather in the sense of building a *computational* model and implementing the corresponding data structures and algorithms of recomputing beliefs. The purpose of such a project is to illustrate some ideas about belief changes in a Web of Beliefs (WoB) to explore and deepen one's understanding of belief changes by trying to implement or improve corresponding algorithms. With implementation one will proceed from more basic models, which make a lot of simplifications, to more extended models, which look more at internal belief structure and employ more refined algorithms.

## **§1 Belief Dynamics in a Web of Beliefs**

Belief dynamics involves all ways of updating or changing a belief system, Web of Beliefs. This may be expansion, contraction or revision. Partial analyses have been developed by the AGM-approach, and other authors of the field of 'belief revision', and other (AI)-authors dealing with inductive and non-monotonic logic, dynamic epistemic logic and corresponding parts of epistemology. One may see philosophy's role in the cognitive sciences here again as *inter alia* providing (formal) models of basic entities and capacities. This formal modelling is continuous to the formal modelling in some cognitive sciences, as (formal) linguistics or 'ontologies' in Artificial Intelligence (there being only a vague boundary between work in Artificial Intelligence and research in pure formal logic). Given the sheer number of researchers working in the fields, however, much more detailed work in formal modelling is done outside of philosophy and more in the cognitive sciences, especially in abstract computer science and the related fields of logic. Some areas like 'knowledge representation' and also belief dynamics fall squarely in the area of traditional epistemology. So, again, the role of philosophy cannot be to supply all these models and results, but to consider there most general aspects and components, and to provide some theoretical general framework wherein to place these approaches.

A Web of Beliefs is a graph/network with beliefs as nodes and directional inks of support by other beliefs and supporting other beliefs. The support may be deductive or inductive (in the broad sense, including defaults and direct support by testimony or other sources). A WoB does not just include the beliefs held (the kernel of the WoB, the 'Belief Box'), but also other known sentences of the agent's language, which may be adopted at some point. An agent knows these 'possible beliefs' as options, and can reason about the effect of adopting one of them (e.g. thereby making a premise of an argument true which supports another belief). Surrendered beliefs are

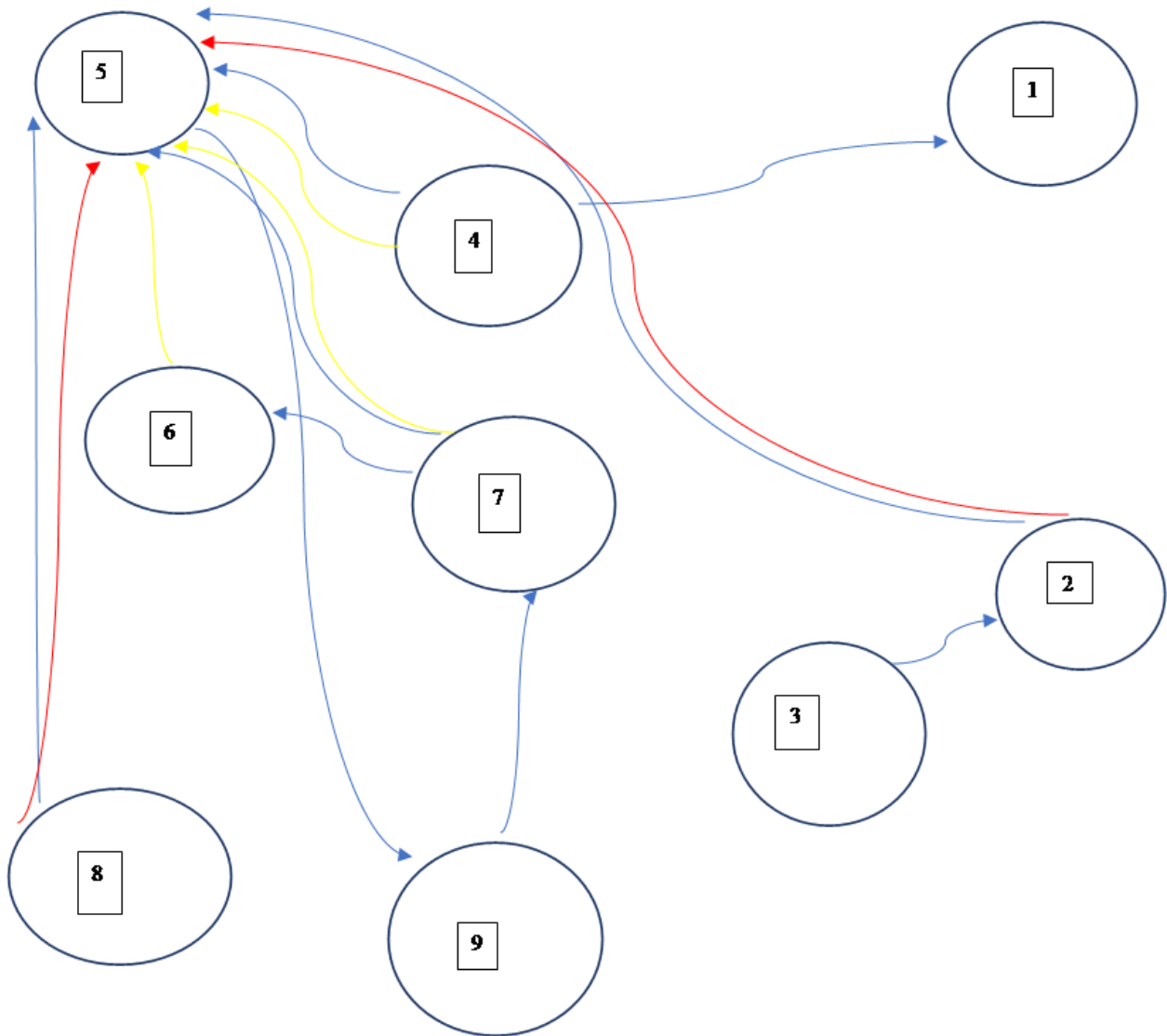
not excluded from a WoB (or forgotten), but can be re-adopted.

Theories like AGM express principles that define the transformations of 'expansion', 'contraction', and 'revision'. These meta-theoretical principles (expressed using set theory and notions of derivability) state pre and post conditions of the transformations the workings of which are a black box. Computational approaches try to specify the algorithms within that black box. Principles of theories of dynamic epistemology can then be considered as constraints what these algorithms should deliver.

Belief dynamics consists in cycles of re-computation (CRC). The focus here is on the computational procedures of belief changes (as pioneered by Neil Tennant's project of *Changes of Mind* and John Pollock's Oscars project of *Cognitive Carpentry*). The focus is on *cognitively penetrable procedures* of belief dynamics involving changes in justifications and links between beliefs, which distinguishes this approach from probabilistic approaches. In this perspective belief dynamics consists in maintaining justificatory closure and justificatory consistency conditions (as distinguished from both objective deductive closure and probabilistic conditionalization).

See an illustration of a Web of Belief here:

---



**This crude picture tries to illustrate a Web of Beliefs. The nodes are beliefs and they have directional links (with a base) to other beliefs. The blue lines depict a supporting relation (e.g. from 3 to 2). The red lines depict support by a group of beliefs, also the yellow lines (e.g. “A”, “If A then B” support as a group {2,8} “B” {5}, which is additionally supported by {4,6,7}). In a Web of Beliefs as a graph there may be also circles of support (e.g. between 9 and 5). Graph transitions thus require track keeping.**

## §2 Logical and Psychological Models

We should distinguish a logical model of a belief system and the accessible belief system. The logical model can contain data structures and information that is not consciously accessible to the agent. *Conscious access* has to play a part as belief dynamics is part of deliberation and conscious and not only sub-conscious 'online' interaction with reality. The agent does not simply forget any beliefs. This idealization could be removed by randomly forgetting (i.e. removing) some beliefs, but such a random approach seems to be not better justified than the idealization. The focus of the model lays on principles of rational belief changes, not on a complete psychological account of how a Web of Beliefs develops over time in persons. No severe idealizations should be made with respect to logical knowledge.

The logical model contains the links between the nodes. A psychological model may reckon with the psychological fact that we cannot easily access these links introspectively. Thus the psychological model may rather look like a set of nodes (containing sentences believed). CRCs and BD depend on (partially doxastic and partially sub-doxastic) access to links. Positive undermining may deactivate a belief directly. Loosing justification requires some track keeping of the links towards a belief.

Inquiring into an issue may involve either (i) accessing previously established and still present links in a (logical) graph of beliefs, *or* (ii) constructing routes to evidence for the issue in questions starting from a semantic parsing of the sentences expressing the issue. In case (ii) the links may either (iia) be more easily be remembered by this prompting, but still preexisting or (iib) constructed on the spot by accessing the repository of other beliefs in a belief set. A repetitive reconstruction of links seems a waste of resources. Online total recall of justifications seems a waste of conscious awareness and may result in a loss of focus. Semantic prompting

of analytic and justificatory links may be more economical than a facility of total recall on the spot.

### **§3 Justifications in a Web of Beliefs**

A computational approach to issues of belief dynamics, i.e. algorithms and a data formats, proceeds from 'crude' to 'refined'. 'Crude' beginnings try to get a grasp on modelling BD, neglecting issues of storage waste, time complexity and at least some questions of psychological reality. 'Refinement' then tries to proceed to models and algorithms that meet the goal of providing a feasible model.

The web of beliefs can be modelled in a graph. The nodes of the graph are sentences. They are expressed in a First Order Language. The links between sentences are annotated. Each node can have finitely many links to other nodes. Some nodes are basic nodes (e.g. sensory input). Basic nodes depend inferentially only on themselves. Whether they imply e.g. perceptual beliefs, however, may depend on normality assumptions etc. Another type of basic nodes are 'postulates' (like axioms of mathematics). Postulates are not confined to logical truths, but they are treated as those sentences not to be given up, and thus are 'valid' as far as the WoB under consideration is concerned.

Some sentence A can be linked to some other sentence B by being part of a premise set that allows to derive B. Sentences are marked for supporting other beliefs, e.g. by assigning them a set of supported beliefs. One measure of the entrenchment of a belief could be the size of this set. To avoid trivialization by the fact that any sentences may have infinitely many logical consequences (these being supported sentences) the entrenchment level should (i) only measure those justifications (by increasing entrenchment one degree) in which the sentences has been used as premise

in addition to at least one more sentence (i.e. has been a relevant premise), and (ii) neglect irrelevant consequences (like those by the rule of Disjunction Introduction). At least some logical truths need no level of entrenchment as they are consequences of the logical framework of rules of derivation, and thus are unrevisable anyway (e.g. conditionals expressing basic rules of detachment).

One may say that the direction of support is 'downwards' (spreading belief downwards) and the direction of being supported is 'upwards' (depending on those beliefs).

Justifications are introduced according to the supposed knowledge of the agent (i.e. they may be objectively conclusive or not). Such justifications or support sets of premises contain only the premises needed (i.e. contain neither irrelevant further premises nor logical truths). Justifications involving the falsity of some claim/sentence use the negated sentence. Justifications express the agent's argumentative power and command of justificatory resources. From an objective point of view they can be quite mistaken. Still an agent can be rational in revising a WoB which incorporates bad arguments.

As justifications can be added to a sentence without all the premises being true they might be distinguished explicitly as operational or not. As an agent may also consider justifications, say heard about, without accepting them, the WoB may contain them, marked as not endorsed. They may be endorsed at some point, leading to an expansion. Contractions work only on the operative justifications. They will not add justifications or make them operational.

The link matrix contains sets of sets of sentences. All these sets are finite as we are dealing with finite WoB. The matrix entry indexed by sentences A and B contains all sets of premises such that A together with these premises logically implies B. These sets of sets are the annotation of the links of the

graph of beliefs (beliefs expressed as sentences). Sentence A can be linked to B by more than one minimal set because A may be part of quite different arguments, even differing in supposed 'logic' (i.e. one being inductive the other deductive). A WoB has sentences as nodes (and no other type of nodes), but differs from a simple directed graph by: the (possible) presence of different types of edges (say deductive or inductive); the collective character of a set of nodes linking to some other node; the presence of multiple edges coming from or arriving at a node (i.e. a node may participate in different sets of different types of collectively linking to a target node).

Surrendering a belief requires invalidating all justifications for it (i.e. surrendering at least one belief in all of them, which one might be random in a basic model or work on their comparative entrenchment in the WoB).

Justifications in which not all premises are held beliefs need not be given up, as the cogency of the justification need not be questioned by the agent, but once all of them supporting some belief have a missing premise the supported belief has to be surrendered.

Adding a justification for a sentence may lead to adopting a belief not held formerly because the premises of this justification are held true. An expansion check may look for any beliefs that should be adopted but have not been so far. Adding a justification should trigger this check. Both surrendering and adopting trigger procedures of justificatory closure.

An agent should also be able to give up justifications (no longer endorsing them), which may lead to a contraction of the WoB. In case there was only one justification for a belief that belief might be surrendered (if not being basic or unrevisable). Changing one's mind with respect to a justification goes beyond surrendering one of its premises. It expresses a changed belief in the acceptability of the justification. As such it might apply to any justification. And if the doubt was centered on the deficiency of the inference



itself, it should apply then to all justifications of the same logical structure! This may be a standard for a fully rational agent. In a model it presupposes that the structure of a justification can be extracted.

#### **§4 Inferences**

The links between sentences are inferential steps in a given logic. By *Hilbert's Thesis* (using the *Church-Turing-Thesis* and Turing's Theorem that First Order Logic (FOL) is equivalent in computational power to Turing-Machines) one can argue that in as much as the inferential rules of some type of reasoning or some (probabilistic, inductive, non-monotonic, adaptive or otherwise non-classical) logic can be specified algorithmically (be it in the meta-language of some calculus) these rules can be rendered as conditionals that have a representation in FOL. FOL is the upper limit of inference. The rules of FOL are finite. The links between nodes represent steps of reasoning. We should assume that steps are minimal in the sense that all premises of a step of reasoning are used in that step. Otherwise everything would be connected to everything else by being a irrelevant/vacuous premise in some step of reasoning.

Belief systems contain finitely many sentences. They are not deductively closed under FOL. Agents are not logically omniscient. An agent may even follow an inference rule of her own logic that is not sound, the corresponding conditional (expressed in FOL) is then one of her beliefs. Belief systems need not be logically consistent, but the drive for at least local justificatory consistency will be part of belief dynamics. As WoB can be inconsistent they are not closed under FOL *ex contradictione quodlibet*.

The algorithms of belief change incorporate a logic of belief change, without which changes will not happen systematically. This logic and these algorithms are, at least in a basic model, not part of the WoB. They are part

of the framework (the agent) treating the WoB as their object. They are not revisable themselves. In an extended model aiming at an universal logic of belief dynamics and the logic of some CRCs itself may be an object of reflection itself. How much of it can be changed - using at least some other logical principles - remains then to be seen.

The rationality of a WoB can be expressed as a set of constraints on beliefs being held true. Re-evaluation of a sentence occurs to fulfil these constraints. Such constraints are: (i) All basic beliefs are held true; (ii) A belief with a support set all elements of which are held true is held true, unless it is not truth-functional and that support set is truth-functional; (iii) A non-basic belief with no support set all elements of which are held true is not held true, unless it is unrevisably held true. These constraints can also be expressed negatively, e.g. (ii') A belief not held true has no support set all elements of which are true, unless ...

What drives belief change? In a consistent setting (i.e. one requiring consistency as a necessary epistemic value) it is avoiding to arrive at an inconsistency. If one allows for inconsistencies at some points, this does not seem to work. Evaluations of sentences, however, even in the presence of contradictions are not arbitrary. And even if we allow for some irrationality, non-truth-functional belief fixation and limited reasoning, still a rational agent aspires to have a WoB as coherent as she can make out. The rational revision of a WoB serves this aim. Its procedural enactment is captured in CRCs.

## **§5 Cycles of Re-Computation**

Belief dynamics consists in cycles of re-computation. Given a change of belief (one node being activated, deactivated or added) or change of links the repercussions of a local change spread stepwise. 'Change' in the broad

sense covers the different kinds of changing the Web of Beliefs as 'contraction', 'expansion', 'switching' etc. The broad category of procedures employed here is called "recomputations". Typically they change truth values of some beliefs and the operational status of some justifications.

There may be a continuous activity of re-computation in the background, and more pressing dynamics in the foreground.

Recomputations could be event (change) driven, but could also be modelled as occurring after regular checks. One may check, for instance, if a belief's status as being believed (considered true) has changed. The set of supported beliefs then needs recomputation. If the change was adopting the belief support has to be spread among the set of sentences supported by that belief (i.e. the last missing premise of a justification for one of them may just have become true) - expanding the WoB. If the change was surrendering the belief supported beliefs have to checked whether they have just lost their last justification - contracting the WoB.

Because of several changes being present in a WoB a CRC working locally in its steps can lead not just to further need of change, but also to revisions of changes in the current list of changes to be handled! A CRC by itself does not guarantee a stable WoB or even a WoB justificatory consistent in its support relations. This needs to be checked time and time again, and handled in further CRCs. Updating by CRCs stepwise in a locally focused manner delays exponential explosion of computation time and resources by allowing the global state of the WoB to be permanently deficient and 'under construction'. One idea behind the concept of 'Cycles of Recomputation' is to focus not on an overarching global algorithm which settles the WoB in a stable state proceeding from another stable state, but on local changes, which then trigger further (synchronized) changes. To avoid mutual undoing of these local processes, and spreading incoherence in the WoB, these local processes have to be synchronized in some fashion. Either a locally started

algorithm has to block changing beliefs in its domain, or a conflict resolution (e.g. by comparing levels of entrenchment) has to resolve the conflict, or one allows for the conflict and assumes that in the course of further CRCs a more stable WoB is reached. An agent committed to adopting some belief but also committed to giving up some other belief may be stuck in such an unstable epistemic situation of reoccurring revisions, which then may be the motivation to re-evaluate some justifications, or to forsake one of the commitments. Such conflicts may be identified by identifying especially unstable beliefs. An instability check may look for beliefs which have been more often re-evaluated than some threshold. As these beliefs can be the source of conflicts of CRCs one safe strategy could be to surrender them all, which should resolve the conflicts. Even for observational and rather quickly fluctuating beliefs one may identify an update problem by computing their ratio of stability to the count of CRCs. If that ratio is too high (indicating that they switch with almost every CRC) there may be an interdependence problem lurking here.

CRCs are connected to algorithms of graph traversal. In each CRC the link matrix is updated. A consistent link matrix is stable [where 'consistency' as always is understood as maximal in the sense that all remaining inconsistencies cannot be avoided given the basic principles of that belief system]. As long as the links matrix is unstable there will be CRCs.

If the truth value of a sentence changes it should initiate that the truth values of its truth-functional dependencies are recomputed. Their value changing may initiate a cascade of further changes. Simple and complex sentences may be supported non truth functionally. The overall state of the Web of Beliefs, therefore, need not be truth functionally consistent. This can be enforced, however, by pushing the proper dependencies downward or computing truth values upward. A single step of recomputation need not result in an intended (e.g. truth functionally consistent) state of the Web of

Belief, so one needs cycles of recomputation. (We may count the number of cycles.)

The graph structure open to cycles of justification and the existence of basic nodes together combine elements of coherentism and foundationalism.

BD does not spell out epistemology in terms of either of these structural claims, it can however be considered to be part of an *elucidation of coherence*. BD models the dynamics of rationality in maintaining a coherent Web of Beliefs. If the ways of connecting beliefs are modelled by the steps (these being explanatory steps or inductive steps or ...) then coherence may come down to two ingredients: (i) from the internal perspective within the WoB coherence is maintenance of justificatory consistency by appropriate CRCs , (ii) from an external perspective one may ask whether the link structure of a consistent WoB  $\mu$  is more coherent than the links structure of a consistent alternative  $\mu'$  where  $\mu$  and  $\mu'$  share (most of) their nodes. Considerations of type (ii) are the classical desiderata of spelling out 'coherence' by 'explanatory power', 'simplicity' etc. An account of BD need not consider (ii) in the beginning, maybe (ii) can be part of an account what triggers changes in links or postulates.

Once a belief is no longer held the supported beliefs may lose their support if their sole justification involves the belief. So, presence of support should be checked regularly. Knowing some justifications and holding their premises to be true should lead to closure of inferential knowledge. This is the goal of a closure check. A 'positive' check ('expansion check') looks for beliefs which should be taken as true and are not, so far. A 'negative' check looks for beliefs being still held true, although they have lost their justification ('contraction check'). If a justification is undermined by giving up one of its premises, this may have repercussions for all justifications in which this premise also occurs. So, a local contraction should trigger further checks for unsupported beliefs.

Surveying changes of beliefs after having initially checked for inferential closure, should diminish the need for closure checks. As the different types of checks work in different directions (expansion or contraction) their parallel execution could result, for example, in a belief the support of which should be made inoperative being reinstated. Which checks are called may depend either on an epistemic schedule or decision of the agent, or a background schedule runs the checks at intervals independently of each other, or the algorithms are synchronized in their access to (sets of) beliefs.

## **§6 Beliefs**

Beliefs are modelled by sentences either held (to be true) or not, in the simplest model. The sentence states the belief's content. An extended model may include further truth values as means of evaluation. In any case the WoB itself does contain both beliefs held and other contents, which may be adopted at some point. Selecting the items held (to be true) provides the 'Belief Box' of the agent, involves, for instance, in practical reasoning.

One may mark beliefs further as 'unrevisable', so that contractions of a WoB will not result in them being surrendered; one may mark them as 'basic' so that they are their own justification (need no further support sets, although they may have them). 'Unrevisable' means 'held come what may', extended models may make also this property revisable. Such basic beliefs that either have no justification, or are their own justification, and beliefs that depend in their justifications only on such basic beliefs are 'safe' beliefs: they cannot be affected by contractions, and thus will never be revised. They constitute background indefeasible belief.

A belief may be qualified as not being truth-functional, which means that the belief may be revised, but not for mere truth-functional consistency.

Commitment to beliefs, thus, comes in qualities and degrees. These play a role in contractions. The degrees are the degrees of entrenchment. A belief may also have the property of 'stability' measuring how often the belief has been re-evaluated (e.g. by counting changes), and carry the marker 'rather stable' to point out beliefs which should not be the object of frequent changes. This, obviously, makes no sense for quickly fluctuating beliefs like observational beliefs (e.g. "The sun is covered by a cloud"), but only for beliefs that are in a sense 'crucially interesting' and thus relevant with respect to change (e.g. "The speed of light is constant."), which typically will be theoretical statements of beliefs of practical or moral importance.

Disbelief may be modelled as believing a negation in contrast to abstaining from belief. In a simple model there may be just held beliefs and sentences not held to be true. Belief is not exhaustive (i.e. one may neither believe A nor not-A). One may, however, believe both A and not-A (at least implicitly). In contrast to not holding a belief one may hold the higher order belief that this content is 'unkown' or 'indeterminate', which is an epistemic belief. In a basic model this may be identified.

All beliefs can be re-evaluated, even logical truths, as agents are not logically omniscient. Their revision may, of course, spread more objective error in the WoB.

## **§7 Algorithms of Belief Change**

Complex sentences can be truth functional in their components if the truth values of the components are known. Webs of Belief are not deductively closed. Reasoners are only finitely interested in consequences and not logically omniscient. So having a belief 'A' does not entail having the beliefs 'A or B', 'If C then A' etc. In cycles of recomputations (even deductive ones) not all consequences are drawn or added to the Web of Belief. Agents are

neither logically omniscient nor perfectly rational. They can even re-evaluate logical truths as not being true. They know the basic - say natural deduction - rules of (propositional) logic. This seems (as these introduction and elimination are phrased) are rather viable assumptions. It does not include logical omniscience in knowing all the consequences of applying these rules (in theorems or in inferring from a set of assumptions). The theoretical model explored here focusses on the rules of recomputation - what rational rules for recomputation there might and should be. These rules can be applied to widely irrational belief sets to update them rationally. Over the long run such belief sets should become more rational. In the computer model the algorithms are employed as they should, real agents may also be faulty with respect to their employment of BD rules. Because of that atomic changes of mind involve simple sentences ('predications') directly and complex sentences indirectly, in a (possible) next computation of revised truth values. A contraction of a Web of Beliefs occurs when a belief so far held to be true is given up (surrendered) by completely removing it from the Web of Belief (by retraction) or assigning it a truth value not implying truth. A revision occurs when switching with respect to the truth value of a belief. So, opt contractions are revisions in a broad sense. Retractions lead to contractions in the narrow sense, as the Web of Belief decreases in size. As the belief in question may have supported other beliefs recomputation of support is called for also in this case. Therefore, in all cases we need algorithms of belief and justificatory changes. An expansion is adopting a belief as true, not formerly held or not considered to be true. An expansion in the narrow sense is adding a new belief to the Web of Beliefs. As a Web of Belief allows for beliefs being present but being unknown in truth value, adopting such a belief as true is an expansion. So, again, we employ algorithms of BD.

A revision in the sense of the 'Levi identity' may be implemented by (i)



switching with respect to a belief, (ii) executing a contraction check, (iii) switching with respect to the negation of the belief, (iv) executing an expansion check.

Contractions should satisfy a constraint of minimal mutilation (not giving up more than needed to surrender a belief in question). The algorithms affect at least and at most as many other beliefs as needed for the revision. In case a belief is no longer believed each support set has to be made inoperational by making just one of its premises false. Support sets need a method returning the least entrenched belief or the first of these if there is a tie in entrenchment levels.

As contractions may be indeterministic (either by selecting one of the premises of a justification of a surrendered belief to be surrendered for contraction, or even in extended models if there happens to be a match of entrenchment for beliefs to be surrendered) a principle of 'recovery' (that revising a contraction by an expansion with the belief in question) is *not* plausible. If justifications are not cut from the WoB in a contraction, they can become operational again. This might re-install a belief formerly surrendered, but the indeterminacy of contraction does not allow to guarantee this.

## **§8 Computability**

Algorithms of BD are - by definition - computable. Therefore they do not coincide in general with postulates on contraction, revision, and expansion made with respect to infinite belief sets, as in AGM-theory or 'Ranking Theory'. Again, the focus of a computational model is on the local logic of revision algorithms (in the broad sense) and on rationality constraints in employing them.

As the CRCs enforce - inter alia - truth-functional dependencies their

complexity must be at least NP-hard, one may suppose. Given that justifications may use non-standard logics and inductive or default reasoning the complexity of decidable parts of revision may, once a model includes tracing such logical structures, be way beyond NP and PSPACE. Such classifications are epistemologically not immediately damaging as algorithms of revision operate locally in cycles of recomputation. Further on, revisions operate on the justifications adopted by an agent giving her own lights, so in a basic model they may look at them as sets of items one of which has to be removed regardless of their structure. This may still be exponentially explosive as re-evaluating one belief forces the algorithm to look at the set of justifications (a set of sets of beliefs), and then follow with assessing their comparative entrenchment, and then forwarding the consequences of an re-evaluation for each of these sets.

The support sets do not contain copies of sentences but only references to existing sentences. Maintaining an array of edges (including enlarging it whenever a belief is added) and collecting support sets from the entries of this matrix may be more computationally involved.

## **§9 Adopting Beliefs**

Adopting a belief means that the belief either is not in the Web of Beliefs so far, and now enters as a belief held to be true, or an existing belief is supplemented with new support. If a complex sentence is adopted as basic this can be done with or without truth functionality intact. Adopting for reasons means adding a support set for a belief each member of which is considered to be true (falsehoods dealt with by negations).

Reasons can be added to beliefs without either holding this support nor the belief to be true (as in seeing this justificational connection). We can know of a justification and endorse its validity without supporting its premises or

conclusion. Giving up a premise or conclusion leads to revision, but this need not involve giving up the knowledge of the justification being available. Some beliefs (say experiential beliefs) depend only few other beliefs (e.g. about the meaning of words used in their description), but even observational beliefs are dependent at least on presumptions of normal conditions, which can be overridden. The majority of beliefs are held conditional on other beliefs: their (implicit) justifications. Belief dynamics occurs in large parts by changes in the status of those conditions on which a belief is held or withheld. Adopting some belief has repercussions to the adoption of other beliefs the justification of which just depended on this belief. Giving up some belief has repercussions as the justifications of other beliefs may be undercut.

Given these dynamics conditionals can in part be understood by different types of 'Ramsey Tests'. Material conditionals and strict conditionals (entailments) have static truth conditions and are used - for most - in regimented scientific theories or (in case of entailments) as expression of analytic connections between concepts. For indicative conditionals their acceptability can be captured by one type of Ramsey Test: If the expansion of a WoB with the antecedent A justifies the adoption of the consequent B, we accept the indicative conditional 'If A then B'. This mirrors one of the paradigmatic ideas of justificationist semantics (in intuitionism or natural deduction): a conditional is justified if a justification of the antecedent can be transformed in a justification of the consequent. For counterfactual conditionals their acceptability can be captured by another type of Ramsey Test: If the revision of a WoB with the antecedent A justifies the adoption of the consequent B, we accept the counterfactual conditional 'If A was the case, B would be'.

## §10 Simplifications in Basic Models

Not all sentences of the language an agent uses are part of her/his Web of Belief, not even all inferentially connected sentences are. Therefore a finite repository of sentences is used to draw sentences from.

In a first model tense should not be not taken into account. Sentences have their truth value at some isolated state. A recomputation/update of the Web of Beliefs leads to a new state of belief (with no tense marking for the transition). Quantification may also not be taken into account in the first model to focus on the revision rules and to leave issues of verification to the side, for now. Some belief content may be considered true or false or unknown, but maybe also considered indeterminate (i.e. not in the sense of epistemologically unknown, but semantically indeterminate) or even considered as contradictory. Thus, five truth values may be taken into account in an extended model, or at least two in case of bivalence.

---

A basic model of algorithms of belief dynamics implemented in JAVA:  
The central class WebOfBeliefs (containing the algorithms) uses the classes Sentence and Justification.

---

```
import java.util.HashSet;
import java.io.Serializable;

public class Sentence implements Cloneable, Serializable
{
    private boolean believed = false;
    private final String content;
    private int entrenchment = 0;
    private HashSet<Justification> supportedBy;
    private HashSet<Sentence> supporting;
```

```

/**
 * Constructor for objects of class Sentence
 * @param - the sentence's content as a String (proper content is not evaluated)
 */
public Sentence(String content)
{
    this.content = content;
    supportedBy = new HashSet<Justification>();
    supporting = new HashSet<Sentence>();
}

/**
 * Adds a (further) justification for this sentence.
 */
public void addSupport(Justification j){
    supportedBy.add(j);
}

public void addSupportedBelief(Sentence s){
    supporting.add(s);
    entrenchment++;
}

@Override
public Object clone() throws CloneNotSupportedException{
    Sentence clone = new Sentence(content);
    if(believed){
        clone.endorse();
    }
    clone.setJustifications(supportedBy);
    clone.setSupported(supporting);
    clone.resetEntrenchment(entrenchment);
    return clone;
}

public Sentence copy(){
    try{
        Sentence copy = (Sentence) this.clone();
        return copy;
    }
    catch(Exception e){
        System.err.println(e);
    }
    return null;
}

public void endorse(){
    believed = true;
}

public int entrenchmentLevel(){
    return entrenchment;
}

```

```

public String getContent(){
    return content;
}

public HashSet<Justification> giveJustifications(){
    return supportedBy;
}

public HashSet<Sentence> giveSupportedSentences(){
    return supporting;
}

public boolean isBelieved(){
    return believed;
}

/**
 * @return - the negation of the sentence; applying double negation elimination.
 */
public Sentence negation(){
    String negatedContent = "";
    if(content.startsWith("It is not")){
        negatedContent = content.substring(24); // the part after the negation
    }
    else{
        negatedContent = "It is not the case that " + content;
    }
    return new Sentence(negatedContent);
}

public void resetEntrenchment(int level){
    entrenchment = level;
}

public void setJustifications(HashSet<Justification> support){
    supportedBy = support;
}

public void setSupported(HashSet<Sentence> supporting){
    this.supporting = supporting;
}

public void surrender(){
    believed = false;
}

@Override
public String toString(){
    return "[" + content + "]" +
        ((isBelieved())? "is believed" : "is not believed") + ".";
}
}

```

---

```

import java.util.HashSet;

/**
 * Justification - a justification for a belief.
 * The justification may be operative (endorsed at that point) or not.
 * A justification consists of premises/sentences which in the eye of
 * the agent support a conclusion in question.
 */

public class Justification
{
    private boolean operative;
    private final HashSet<Sentence> premises;
    /**
     * Constructor for objects of class Justification.
     * A justification may consist of any number of premises.
     */
    public Justification(Sentence... args)
    {
        premises = new HashSet<Sentence>();
        for(int i = 0; i < args.length; i++){
            premises.add(args[i]);
        }
    }

    /**
     * Sets the justification operative.
     */
    public void endorse(){
        operative = true;
    }

    /**
     In case a justification has to be givenup, because its conclusion is surrendered,
     the least entrenched premise has to be found, to be surrendered in a contraction.
     */
    public Sentence giveLeastEntrenched(){
        Sentence out = new Sentence("No Content");
        int minimum = Integer.MAX_VALUE;
        for(Sentence s: premises){
            if(s.entrenchmentLevel() < minimum){
                out = s;
                minimum = s.entrenchmentLevel();
            }
        }
        return out;
    }

    public HashSet<Sentence> givePremises(){
        return premises;
    }
}

```

```

    public boolean isOperative(){
        return operative;
    }

    /**
     * Sets the justification inoperative.
     */
    public void surrender(){
        operative = false;
    }
}

```

---

```

import java.io.Serializable;
import java.util.HashSet;
import java.util.LinkedList; // implementing a queue
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;

```

```

/**
 WebOfBeliefs - a basic WebOfBeliefs with:
 1. unstructured sentences
 2. containing as nodes a subset of the sentences of a language (believed or not)
 3. containing with each sentence its negation as well (in case needed for a revision)
 4. nodes are never cut, beliefs may be surrendered (disbelieved), but still present
 5. in expansions nodes are not added but switched from disbelieved to being believed
 6. a static structure of nodes and their justifications (operative or not);

```

*Algorithms of belief revision work globally (i.e. through the whole WebOfBeliefs). They work deterministically and without foresight (to a better choice of surrendering beliefs), so, although Contraction looks at entrenchment of beliefs, the changes may not be minimal.*

```

*/

public class WebOfBeliefs implements Serializable, Cloneable
{
    private HashSet<Sentence> beliefs;
    private final int identifier;
    private int size = 0;

```



```

/**
 * Constructor for objects of class WebOfBeliefs
 * @param - an identifier (e.g. for the file name if saved in a file)
 */
public WebOfBeliefs(int identifier)
{
    beliefs = new HashSet<Sentence>();
    this.identifier = identifier;
}

/**
 * Beliefs are always added with their negation as well, as one may need the
 * negation for revision. Adding a belief does not mean endorsing a belief.
 * If the added sentence comes with justifications their premises have to be
 * added to the WebOfBeliefs as well. The Justifications need not be operative.
 */
public void addBelief(Sentence s){
    beliefs.add(s);
    beliefs.add(s.negation());
    size += 2;
    for(Justification j: s.giveJustifications()){
        for(Sentence premise: j.givePremises()){
            if(!beliefs.contains(premise)){
                addBelief(premise); // this may pass on further additions
            }
        }
    }
}

@Override
public Object clone() throws CloneNotSupportedException{
    WebOfBeliefs clone = new WebOfBeliefs(identifier);
    for(Sentence s: beliefs){
        // copies of the beliefs are used to not interfere with further processing
        clone.addBelief(s.copy());
    }
    return clone;
}

/**
 * Global Contraction Algorithm
 * Given a belief to be surrendered, the support has to be undermined
 * (following the links to justifications of this belief).
 * Additional to this upward direction of undermining, Contraction
 * has also to pass the new status of the surrendered beliefs to the
 * beliefs its supports in downward direction as these may have
 * to be surrendered as well.
 * @param - the belief to be surrendered
 * @return - all beliefs surrendered in the contraction
 */
public LinkedList<Sentence> contraction(Sentence s){
    s.surrender();
    LinkedList<Sentence> queue = new LinkedList<Sentence>();
    LinkedList<Sentence> surrendered = new LinkedList<Sentence>();

```

```

queue.addLast(s);
surrendered.addLast(s);
while(!queue.isEmpty()){
    Sentence focus = queue.remove(); // head of the queue
    // upwards
    // we have to look at the justification(s) of the belief given up
    for(Justification j: focus.giveJustifications()){
        // only operative justifications have to be surrendered
        if(j.isOperative()){
            // premise to be surrendered is the least entrenched
            Sentence leastEntrenched = j.giveLeastEntrenched();
            leastEntrenched.surrender();
            // this has to be passed on for further contraction
            queue.addLast(leastEntrenched);
            surrendered.addLast(leastEntrenched);
            System.out.println(leastEntrenched);
        }
    }
    // downwards
    // we look at the supported sentences
    for(Sentence supported: focus.giveSupportedSentences()){
        // sentences not believed are irrelevant for contraction
        if(!supported.isBelieved()){
            break;
        }
        // and look, whether all their justifications may become inoperative
        for(Justification j: supported.giveJustifications()){
            // inoperative justifications need not be contracted
            if(j.isOperative()){
                // crucial are justifications with the sentence surrendered
                if(j.givePremises().contains(focus)){
                    j.surrender();
                }
            }
        }
        // Now we look, whether all justifications have been surrendered
        boolean allJustificationsInoperative = true;
        for(Justification j: supported.giveJustifications()){
            if(j.isOperative()){
                allJustificationsInoperative = false;
                break; // one justification is enough
            }
        }
        if(allJustificationsInoperative){
            supported.surrender();
            queue.addLast(supported); // contraction has to be passed on
            surrendered.addLast(supported);
            System.out.println(supported);
        }
    }
}
return surrendered;
}

```

```

public WebOfBeliefs copy(){
    try{
        WebOfBeliefs copy = (WebOfBeliefs) this.clone();
        return copy;
    }
    catch(Exception e){
        System.err.println(e);
    }
    return null;
}

/**
 * Global Expansion Algorithm
 * Support is passed down the supporting links to further beliefs.
 * @param - the sentence now to be believed
 * @return - the new beliefs generated by the expansion
 */
public LinkedList<Sentence> expansion(Sentence s){
    s.endorse();
    LinkedList<Sentence> queue = new LinkedList<Sentence>();
    LinkedList<Sentence> newBeliefs = new LinkedList<Sentence>();
    queue.addLast(s);
    newBeliefs.addLast(s);
    while(!queue.isEmpty()){
        Sentence focus = queue.remove(); // get head of the queue
        // we look at the supported sentences
        for(Sentence supported: focus.giveSupportedSentences()){
            // beliefs already held need not to be considered
            if(supported.isBelieved()){
                break;
            }
            // look at justifications, whether one may become operative
            for(Justification j: supported.giveJustifications()){
                boolean allPremisesTrue = true;
                for(Sentence p: j.givePremises()){
                    // if one premise is not endorsed,
                    // this justification is not operative
                    if(!p.isBelieved()){
                        allPremisesTrue = false;
                        break;
                    }
                }
                if(allPremisesTrue){
                    // the justification still is or may become operative
                    j.endorse();
                    if(!supported.isBelieved()){
                        // this sentence should now be believed: expansion
                        supported.endorse();
                        newBeliefs.addLast(supported);
                        System.out.println(supported);
                        // the expansion has to be passed on
                        queue.addLast(supported);
                    }
                }
            }
        }
    }
}

```

```

        // we cannot break here as more than one justification
        // can become operative
    }
}
}
return newBeliefs;
}
}

/**
 * Revision requires finding the negation of a sentence (by linear search).
 */
private Sentence findNegation(Sentence s){
    String negatedContent = s.negation().getContent();
    for(Sentence belief: beliefs){
        if(belief.getContent().equals(negatedContent)){
            return belief;
        }
    }
    return null;
}

/**
 * @return - all beliefs, endorsed or not
 */
public HashSet<Sentence> giveBeliefs(){
    return beliefs;
}

/**
 * @return - returns only the beliefs endorsed; the agent's 'belief box'
 */
public HashSet<Sentence> giveHeldBeliefs(){
    HashSet<Sentence> held = new HashSet<Sentence>();
    for(Sentence s: beliefs){
        if(s.isBelieved()){
            held.add(s);
        }
    }
    return held;
}

/**
 * @return - String describing held beliefs and their operative justifications
 */
public String heldBeliefsAndArguments(){
    int arguments = 1;
    String out = "WebOfBeliefs " + identifier + "\nStructure and content:\n";
    for(Sentence s: beliefs){
        if(s.isBelieved()){
            out += s + "\n\t held because:\n ";
            for(Justification j: s.giveJustifications()){
                if(j.isOperative()){
                    out += "Argument " + (arguments++) + "\n";
                    for(Sentence p: j.givePremises()){
                        out += p + "\n";
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
    }
    }
    return out;
}

public int numberOfBeliefs(){
    return size;
}

public void print(){
    System.out.println(this);
}

public void printHeldBeliefsAndArguments(){
    System.out.println(heldBeliefsAndArguments());
}

public boolean restoreFromFile(String fileName){
    try{
        FileInputStream fi = new FileInputStream(new File(fileName));
        ObjectInputStream oi = new ObjectInputStream(fi);
        WebOfBeliefs web = (WebOfBeliefs) oi.readObject();
        oi.close();
        fi.close();
        beliefs = web.giveBeliefs();
        return true;
    }
    catch (FileNotFoundException e) {
        System.err.println("File not found");
    }
    catch (IOException e) {
        System.err.println("Error initializing stream");
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

/**
 * A Global Revision Algorithm
 * Follows the 'Levi Identity'
 * @param - the belief to be switched
 */
public boolean revision(Sentence s){
    // if somethings goes wrong during execution
    WebOfBeliefs copyToRestore = this.copy();
    // Contraction
    if (contraction(s).size() > 0){
        // Contraction has (supposedly) succeeded, we now find the negation
        Sentence negation = findNegation(s);
        if(negation==null){
            // if the negation cannot be found, revision cannot proceed
            System.err.println("Negation could not be found!");
        }
    }
}

```

```

        beliefs = copyToRestore.giveBeliefs();
        System.err.println("Old state of WebOfBeliefs restored.");
        return false;
    }
    else{ // Expansion
        expansion(negation);
        return true;
    }
}
return false;
}
}

/**
 * @return - String describing the whole WebOfBeliefs
 */
@Override
public String toString(){
    int arguments = 1;
    String out = "WebOfBeliefs " + identifier + "\nStructure and content:\n";
    for(Sentence s: beliefs){
        out += s + "\n\t might be held because:\n ";
        for(Justification j: s.giveJustifications()){
            out += "Argument " + (arguments++) + "\n";
            for(Sentence p: j.givePremises()){
                out += p + "\n";
            }
        }
    }
    return out;
}

public boolean writeToFile(){
    try{
        FileOutputStream f =
            new FileOutputStream(new File("WebOfBeliefs" + identifier + ".txt"));
        ObjectOutputStream o = new ObjectOutputStream(f);
        o.writeObject(this);
        o.close();
        f.close();
        return true;
    }
    catch (FileNotFoundException e) {
        System.err.println("File not found");
    }
    catch (IOException e) {
        System.err.println("Error initializing stream");
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}
}
}

```