Theory-Contraction is NP-Complete

NEIL TENNANT, *Department of Philosophy, 230 North Oval Mall, The Ohio State University, Columbus, Ohio 43210, E-mail: tennant.9@osu.edu*

Abstract

I investigate the problem of contracting a dependency-network with respect to any of its nodes. The resulting contraction must not contain the node in question, but must also be a *minimal mutilation* of the original network. Identifying successful and minimally mutilating contractions of dependency-networks is non-trivial, especially when non-well-founded networks are to be taken into account. I prove that the contraction problem is NP-complete.[1]

1 Caveat

The reader needs to know at the very outset what this paper sets out to do, and what it does not. I treat here of finite belief-systems. My interest is in how a finite belief-system can be contracted with respect to any belief in it—how, that is to say, one can give the belief up while at the same time inflicting minimal mutilation on the original belief-system. More precisely, I am interested in determining the computational complexity of the following decision problem: given a finite belief-system T, and any belief p therein, is there a contraction of T with respect to p of such-and-such finite size (less than that of T)?

This paper has only one theoretical aim. I aim to show that the foregoing decision problem is NP-complete. To that end, I need to provide the formal definitions that are required both for the formulation of the problem and for rigorous proof of the claim that the problem is NP-complete. I have no other theoretical aims here than this one. The aim is important enough on its own. It is well known that there are thousands of NP-complete decision problems. But that fact does not entitle one to dismiss the main result of this paper as 'just another NP-completeness theorem'. That would be to underestimate the important of this particular decision problem for the logic of theory dynamics and for epistemology at large. This anticipatory

[1]The main ideas of this paper were first presented to the Interdisciplinary Research Seminar on Mechanization of Inference at The Ohio State University in Autumn Term 1996. More recent versions have been presented to the Ohio State Re-usable Software Research Group; the University of Colorado Colloquium on History and Philosophy of Science; and the Fourth In-House Workshop of the Center for Philosophy of Science at the University of Pittsburgh. Thanks are due to the members of those audiences for comments that have helped clarify the presentation. The Seminar on Mechanization of Inference was funded by a generous grant from the OSU Office of Research, and by grants from the Department of Philosophy and the Center for Cognitive Science. OSU has given me further support in the form of a Special Research Assignment for Spring Term 2003. I am grateful to Victor Marek and Harvey Friedman for helpful discussions in connection with this topic. I would also like to thank Sam Buss, for a careful reading, helpful observations, and an important simplification of my original proof of Theorem 1; and Anish Arora, for the central insight in the proof of Theorem 2. It was Anish who first encouraged me to isolate just the result on NP-completeness for publication as a theoretical piece. In the event, the paper had to expand beyond its original formal limits, in order to meet referees' demands for more informal explanation of the motivation behind my modelling. Hence the current expository texture of the paper, which I hope the general reader will find helpful. Any defects that remain are entirely the author's responsibility.

defence depends, of course, on the formal modelling being accepted as a good first approximation of the epistemologically important phenomena. But arguing in detail that it should be so accepted lies beyond the scope of this paper. So too does the proof of a certain metatheorem, due to Harvey Friedman, that ensures that one incurs no loss of theoretical generality by restricting oneself to *finite* belief-systems.

Despite my focus on proving the main theoretical result about NP-completeness, I am very concerned to make the ideas behind my modelling, and that result, easily accessible to the reader who is not a specialist in the area of belief-revision. For the sake of clarity, therefore, I provide illustrations of defined notions along the way. I also make some comments, which I keep to a minimum, about the motivation behind my modelling of belief-systems and behind the formal definitions that give it expression. These comments are expository only, intended to help the reader who is unfamiliar with the general area. They are not intended as a remotely adequate substitute for the kind of discussion that would be required in order both to justify the motivation behind my chosen form of modelling and to show that it has been given ideal expression in the formal definitions chosen. Such a discussion would need to deal with a host of philosophical, epistemological, logical and mathematical issues. These issues simply lie beyond the scope of this paper. I have already, to some extent, discussed these issues elsewhere (see [16], [17] and [18]) and I intend to discuss them in even greater detail in future publications.

I ought also to give the reader yet further advice as to what this paper is not. It is not the kind of paper that specifies any algorithms. Nor is it the kind of paper that reports on implementations of algorithms. Both the specification and the implementation of algorithms for contracting belief-systems are discussed in detail in other papers that I have already written, and which I intended to submit for publication only after this purely theoretical paper had been accepted. For one needs to know that the computational problem is NP-complete in order to be able to assess any proposed algorithms for solving it. The actual algorithm that I do propose is a development (in successively less greedy versions) of what I called the 'staining algorithm' in [16].

Nor am I, in this paper, concerned to provide detailed critiques of rival accounts of belief-revision and/or contraction, such as *AGM*-theory or JTMS-theory. That, too, has already been undertaken elsewhere (see [16] and [18]), and is to be developed further in future papers. In this paper I confine myself to making the briefest comments possible on both *AGM*-theory and JTMS-theory, simply so that the reader can appreciate how the present approach differs from those.

2 Introduction

The task of belief-revision can be stated as follows. Suppose one is an ideally rational agent, and one's current system of belief implies the proposition p. Suppose that p has just been falsified. Thus one wishes to believe not-p rather than p, and to mutilate one's belief-system as little as possible in coming to do so. First, one contracts the belief-system with respect to p—giving up p and anything in the system that either implies, or wholly depends on, p. Then one adds not-p to the contracted result.

There should be a rigorous account of this operation of contraction. Epistemologists and philosophers of science talk of 'minimally mutilating changes in the web of belief', but give no constructive account of how to effect them. Hence we face the challenge

of giving a philosophically sound, mathematically precise account of contraction, programmable on a computer. The account will be normative, not descriptive: it will reveal what ought to be done by an ideally rational agent confronted with the need to change its mind. Automation of the task would bring widespread applications.

The aim in this paper is to set out an abstract and completely general framework for the discussion of systems of belief and their contractions; to explicate what exactly is meant by minimal mutilation; to characterize the contraction-problem precisely; and to show that this problem is NP-complete.

3 Historical Background

There are two main approaches to the problem of theory-contraction. One is from mathematical logic; the other is from artificial intelligence. My approach is distinct from both of these, seeking to avoid different difficulties inherent in each. I devote a subsection to each of these other approaches, in order to highlight the respects in which my approach differs.

3.1 AGM-*theory*

The approach from mathematical logic, so-called AGM-theory, is due to Alchourrón, Gärdenfors and Makinson. (See Gärdenfors [11] for an account of AGM-theory.) AGM-theory treats belief-systems as theories, that is, as logically closed (hence infinite) sets of sentences. When a theory T is contracted with respect to one of its member-sentences p, the resulting contraction $(T - p)$ is required to satisfy certain conditions, which are set out as the postulates of AGM-theory. The AGM-theorist's aim is then to show that certain functions defined on T and p will produce outputs $(T - p)$ satisfying these postulates. These functions are not, in general, computable, because of their infinitistic character and the undecidability of the underlying relations of logical consequence, reference to which is essential for the theoretical constructions involved. Hence AGM-theory does not offer a convincing prospect for computer-implementation of its methods of contraction.

One of the most important postulates of AGM-theory is the postulate of recovery. Recovery is the claim that if one contracts a theory T with respect to p, and subsequently reinstates p, then one will recover all the original consequences of T. This principle can be counterexemplified in situations for which one's pre-theoretical intuitions are compelling. In this connection see Niederée [14], at pp. 323–4; Levi [13], at pp. 134–5; Hansson [12], at pp. 252–3; and Tennant [16], at pp. 870–4. Recovery violates the following basic intuition: If one believes the proposition b only because one believes the proposition a, and one is giving up a, then one will give up b as well.

My account of theory-contraction differs from AGM-theory on three important points: implementability (on which I insist); the postulate of recovery (which I abandon); and the requirement of minimal mutilation (which I ensure). Moreover, the first difference, on implementability, arises because I am not concerned with (necessarily infinite) logical closures; instead, all the objects and structures with which I operate are finite, and all relations among them are effectively decidable. Moreover, there is a principled philosophical and metalogical justification for restricting our treatment to the finite and computable. This rests on a result proved by Harvey Friedman,

to the effect that there are at most finitely many logically minimal sets of axioms implying any theorem, provided only that the theory in question is axiomatized by means of finitely many axioms and finitely many 'normal' axiom-schemes. (Every extant axiomatic theory in mathematics meets this condition.)

3.2 Truth-maintenance systems

My account of theory-contraction also differs significantly from the prevailing paradigm, among AI-theorists, of so-called truth-maintenance systems (TMSs), reason-maintenance systems (RMSs), and justification-based truth-maintenance systems (JTMSs). (These originate from the work of Doyle [7]. For an integrated overview, see Forbus and de Kleer [9].) My account differs from the various kinds of TMS-theorizing in several ways.

First, unlike the JTMS-theorist (see Forbus and de Kleer [9], ch. 7), I take seriously the possibility of coherentist epistemologies according to which beliefs may be held without their being grounded in beliefs of any privileged class (such as observation reports or sense-datum reports). That is to say, I permit the general possibility of a belief within a belief-system having no well-founded pedigree of support terminating on what the JTMS-theorist calls 'enabled assumptions'. I can still accommodate well-founded belief-systems as a special (and obviously very important) case; but my formal theory accommodates non-well-founded belief-systems as well.

Accommodating the non-well-founded case is sufficient to distinguish my approach from JTMS-theory, *even if* some coherentist epistemologists were to quibble that 'abandoning one belief may mean the remaining beliefs no longer cohere and a radically different set of beliefs need to be adopted'. If the coherentist's complaint here is that I am giving her short shrift, I can simply point out that JTMS-theory gives her even shorter shrift. Moreover, the coherentist has yet to provide a really precise explication of exactly what coherentism consists in, in so far as 'local' relations of justificatory support among beliefs are concerned. There is the very real prospect that at least one systematic explication of coherentism would allow for local relations of justificatory support, and would allow, further, for the general possibility described above—namely, that a belief within a belief-system might have no well-founded pedigree of support terminating on initial beliefs that are themselves in no need of any further support. Moreover, on my approach there is also the prospect of providing convincing evidence for the very claim quoted from my imaginary coherentist critic a few lines back.

Secondly, my primary theoretical focus is on inferential *steps* (identified by their immediate premises and conclusions) rather than on *nodes* (which correspond to sentences, be they premises or conclusions, or both). Any computational treatment of a subject-matter is greatly facilitated by the appropriate choice of data-structure for theoretical manipulation and reasoning. I contend that the most appropriate data-structure is that of the step, not the (labelled) node. I shall not immediately expound on this contention, but shall rather let the ensuing development of ideas and techniques bear it out.

The final and most important difference between my approach and all forms of TMS-theory, including JTMS-theory, is that I tackle the most general problem of how to contract a belief-system with respect to *any* belief that the [J]TMS-theorizer

regards as 'in'. I do not confine the operation of contraction to be simply a matter of 'retracting assumptions', to use the terminology of JTMS-theory. The procedure of 'assumption retraction' is the most complex procedure on offer from current JTMS-theorizing. This procedure is fully deterministic. Given a belief-system $\mathcal{B}$, and any one of its enabled assumptions a, the result of retracting a is *uniquely* determined by $\mathcal{B}$ and a. By contrast, the procedure of 'retracting' a *distant consequence* c of the enabled assumptions within a belief-system (equivalently: contracting the beief-system with respect to c) is in general highly non-deterministic. It is this non-deterministic problem that we address here.

4 The modelling: informal explanation of general features

The formal definitions in the following section will be easy to understand if the reader bears in mind the intended modelling.

4.1 Nodes

Nodes—featureless and unstructured—are to represent particular beliefs within the belief-system of a rational agent. They represent, as it were, specific propositions which the agent might or might not believe. Nodes are the basic objects of which we treat. Everything else that I shall be talking about will be built out of nodes by forming hereditarily finite sets.

4.2 Steps

The agent's beliefs will be represented as standing in relations of support made explicit in what I shall call (justificatory) *steps* involving nodes. A step is fully specified by specifying its finitely many *premises* and its *conclusion*.

I impose a simple rational constraint at the outset. Note that no rational agent would ever seek to justify a belief by appealing both to that belief *and yet other beliefs*. This rational reluctance can be subsumed as a special case of the following more general constraint. *No rational belief system may contain two distinct steps with the same conclusion, one of whose premises are premises of the other step.* Put another way: *steps in a rational belief system require all their premises.* If an agent comes to realize, of a given step in her system, that she can reach its conclusion by using some but not all of the premises of that step, then she will discard the weaker step in favor of the stronger step that uses the weaker (reduced) set of premises.

From now on I assume that all belief systems under discussion satisfy the foregoing rational constraint.

A step whose conclusion is distinct from each of its premises is called *transitional*. Although nodes are basic, in that steps are built up out of them, we shall find that steps themselves are the most convenient data-type with which to work. Note that an agent can know of a step without being committed to any of its nodes, as beliefs. But if an agent knows of a step all of whose premise-nodes she believes, then I take her to be committed to believing the conclusion of that step as well.

Given the rational constraint above, any step with more than one premise will be transitional. For steps with exactly one premise, there are two possibilities: those

where the conclusion is distinct from the premise and hence are transitional; and those where the conclusion is identical to the premise and hence are non-transitional.

In the latter case, we acquire a convenient way of representing an *initial* belief: one that is believed outright, in the sense that as far as the agent is concerned, it requires no further support from any other beliefs that the agent happens to hold. I say in such a case that the belief p in question can be thought of as 'depending on itself'. This does *not* mean that the belief p is *a priori*, and therefore unable ever to be repealed. It means only that the agent believes p outright, in the sense that, as far as she is concerned, she would be justified in believing p even if p were to enjoy no justificatory support from any other beliefs within her scheme. This is of course compatible with p also enjoying such support in certain cases. To make vivid the point that initial beliefs are not necessarily *a priori*, I should point out that my modelling of finite belief-systems could confine itself to just the *contingent* beliefs of the agent. In my representation of the agent's beliefs, a belief's 'depending on itself' (i.e., being believed outright) will be registered by means of a single-premise step, whose premise is the very same node as its conclusion. By adopting this convention we attain a certain smoothness in our mathematical treatment. Note, however, that it is compatible with p's being an initial belief that p stand also as the conclusion of some other, transitional, step, all of whose premises the agent believes. Such would be the case, for example, if p were a simple observational belief that the agent had acquired in the usual way, but for which the agent also had theoretical support—by way of, say, prediction within a higher-level theory on the basis of yet other observational beliefs.

Any rational agent who has a system of beliefs might treat some of these only as initial, that is to say, might believe them outright, without offering yet other beliefs in support of them. In all other cases, a belief will have support within the agent's system. That is, the belief in question will stand as the conclusion of at least one transitional step that involves at least one premise, and all of whose premises are in turn believed. Naturally, a belief can be 'over-determined', by enjoying support from many distinct premise-sets, each of which is *sufficient*, from the agent's point of view, to justify the belief in question. This is what makes the contraction problem, in general, so interesting. For, in giving up such a belief p, the agent will have to give up at least one premise within the premise-set of *each* step that has p as its conclusion.

4.3 Closure

In my modelling, a transitional step is understood as being something of which the agent is aware, or which the agent has acknowledged, as rationally compelling. Hence it is a normative requirement that the agent's system of beliefs should be 'closed' with respect to all such steps. One is, in effect, saying to the agent 'Since you yourself acknowledge the step from $x_1, \ldots, x_n$ to y as compelling, and since you believe each of $x_1, \ldots, x_n$, you should also believe y'. Note that such obligations are *agent-relative*, since the nodes are featureless. I am not adverting to the logical structures of the sentences expressing the beliefs $x_1, \ldots, x_n$ and y, and insisting that the agent should recognize the step as logically valid. Rather, it is taken as a given that the agent herself, for whatever reason, has already acknowledged the step in question as compelling. Her obligation is now only that of *taking the step* to y should she ever

come to believe all of $x_1, \ldots, x_n$.

Let us represent the step from $x_1, \ldots, x_n$ to y by means of the notation

$$x_1, \ldots, x_n \mid y.$$

The reader is advised not to confuse my use of '|' here with other widespread uses of that symbol. The latter include the use on which it is read as 'such that' within a set-abstract like '$\{x \mid F(x)\}$'; as well as its use in logic programming, where it means 'given', as in procedural rules of the form 'goal | sub-goal$_1$, ..., sub-goal$_n$'. My use of '|' might look like a single turnstile shorn of its horizontal stroke: to be sure, premises occur to its left, and a conclusion occurs to its right. But what is designated overall is a justificatory *step* (for the agent). No claim of logical deducibility is made or implied by the use of '|'. For, first, '$x_1, \ldots, x_n \mid y$' is a complex *noun phrase*, meaning 'the step whose premises are the nodes $x_1, \ldots, x_n$ and whose conclusion is the node y'. By contrast, '$\varphi_1, \ldots, \varphi_n \vdash \psi$' is a *statement*, to the effect that there is a proof of the sentential conclusion ψ from the sentential premises $\varphi_1, \ldots, \varphi_n$. Secondly, the nodes are *featureless*, and devoid of logical structure. That is not to say, however, that an agent might not know of a certain step because of an actual deductive proof in her possession, leading from the sentential premises corresponding to the premise-nodes of the step in question to the sentential conclusion corresponding to its conclusion-node. It is just to say that when such a step is represented as a transition, not among sentences, but among featureless nodes, such deductive provenance can be, and is, suppressed. This is in the interests of revealing only such justificatory 'macrostructure' within a belief-system as is relevant for the purposes of contraction. This structure resides at a grosser and more abstract level than that of the detailed logical forms of sentences themselves.

In the course of the agent's investigations, she can come to accept or acknowledge many transitional steps of the form just discussed—

$x_1, \ldots, x_n \mid y$

—while yet failing to believe all of $x_1, \ldots, x_n$. The belief y might fail to enjoy any support, within the agent's belief-system, from sufficient sets of beliefs (from the agent's point of view). In such situations the known steps with y as conclusion are not useless. For, if ever the agent were to come to believe all the premises of any such step, its conclusion y would thereby acquire rational support (from her point of view). This could happen, for example, if the agent at first believed all of $x_1, \ldots, x_{n-1}$ but did not believe x_n; and subsequently came to believe x_n. At that point, since the step

$x_1, \ldots, x_n \mid y$

is known to the agent, she would be rationally obliged to adopt the belief y. That is, she would have to close her beliefs under the step in question.

Clearly, such obligatory operations are iterable, with respect to all known steps all of whose premises have come to be believed. But the iterations will only ever be finite in number. This is because only finitely many steps are known to the agent, and each step involves only finitely many premises. The need for these iterable operations arises only when the agent adopts one or more new beliefs.

Now it is important not to fall prey to the thought that since the agent may know of schematic rules which correspond to an infinity of steps, her belief-system must be infinite. I reiterate the point that my nodes are featureless, and their labels are not to be thought of as endowed with any essentially logical structure. Each node represents a particular belief, as given by a particular proposition, expressible by a particular sentence. No agent will ever have justified to herself more than finitely many such beliefs. Nor will she ever have performed, in her thinking, more than finitely many steps. Indeed, at no time in the future of humankind will the *communal* belief-system, even assuming perfect memory, ever contain more than finitely many beliefs, or involve more than finitely many steps. I call this the *finitary predicament*. Fortunately, it is what makes a computational account of contraction and revision possible! To put the matter succinctly: for the purposes of belief-representation, *schemata are irrelevant; only their instances count*. And at any point in time, there will have been only finitely many instantiations.

I shall call the outcome of the iterable operations described above the *universal closure* of the agent's belief-system, within the set of steps known to the agent, and with respect to the agent's set of newly-adopted beliefs. I am assuming that in this context there is no possibility of confusing this kind of universal closure with the logician's syntactic notion of the universal closure of a formula with free variables, which closure is obtained by prefixing the formula with universal quantifiers to bind its free variables. Universal Closure is called 'propagate inness' by JTMS-theorists; see Forbus and de Kleer [9].

The situation can be schematized as follows. Let $\mathcal{C}$ be the finite set of all transitional steps known to the agent. Every transitional step in the agent's current belief-system $\mathcal{B}$ (which itself is finite) will be in $\mathcal{C}$; but in general there could be (transitional) steps in $\mathcal{C}$ that are not in $\mathcal{B}$. Moreover, $\mathcal{B}$ can contain *initial* steps, none of which is in $\mathcal{C}$. Note that in the belief-system $\mathcal{B}$, every node of every step is believed, whether it be a premise or a conclusion of the step in question.

By way of illustration, take the current belief-system $\mathcal{B}$ to consist of the steps

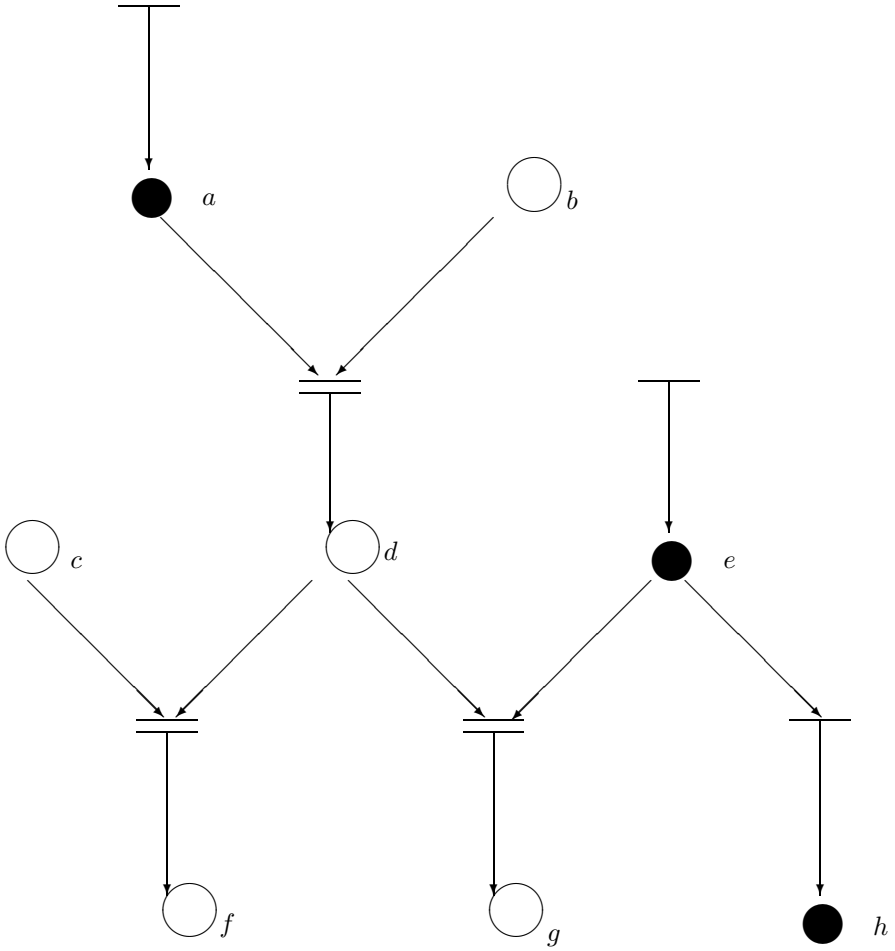
$a \,|\, a$ (initial)
$e \,|\, e$ (initial)
$e \,|\, h$ (transitional)

and let the set $\mathcal{C}$ of transitional steps known to the agent be

$e \,|\, h$ (note: this is also in $\mathcal{B}$)
$a, b \,|\, d$
$c, d \,|\, f$
$d, e \,|\, g$

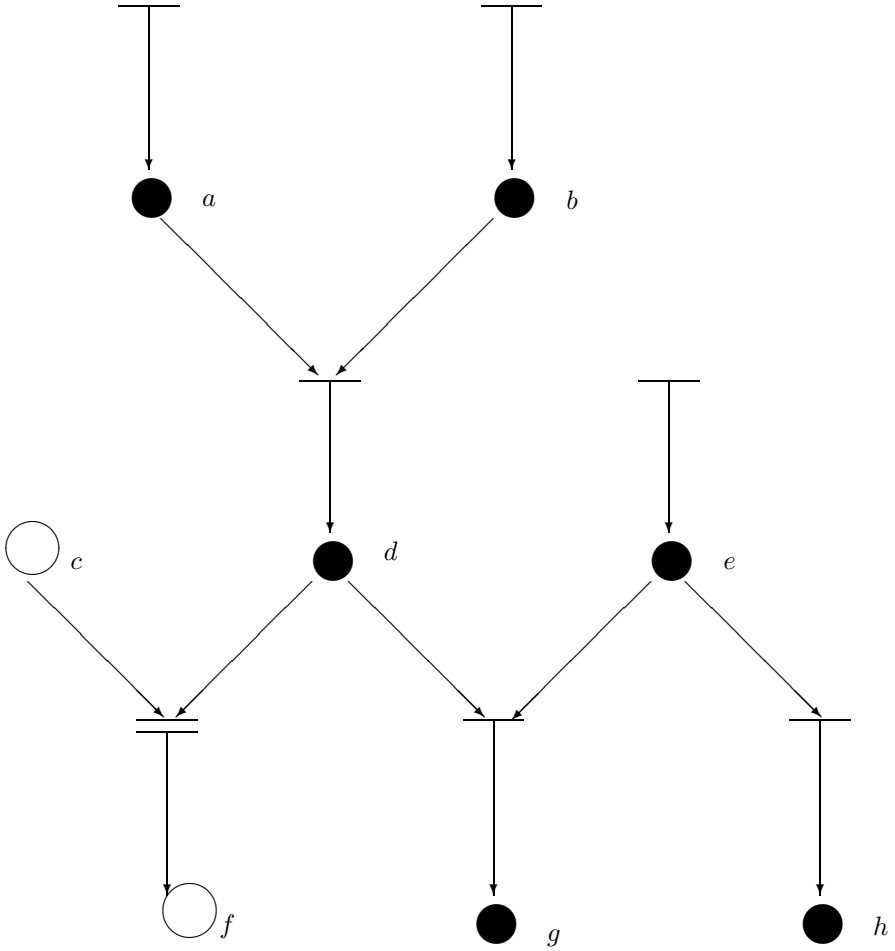
The following diagram represents the combination of $\mathcal{B}$ and $\mathcal{C}$. The black nodes represent the actual beliefs of the agent. These are conclusions of steps in $\mathcal{B}$. The white nodes represent propositions that are not believed, but which feature in transitional steps in $\mathcal{C} \setminus \mathcal{B}$. The initial beliefs (which are all in $\mathcal{B}$) are each furnished with an 'initial arrow' emanating from a horizontal stroke that itself receives no arrows. Horizontal strokes represent the inferential transitions made in steps. The inferential direction is downwards. In general, more than one arrow can come down to a single inference stroke, from the premises of the step in question; while only one arrow descends from

the stroke, to the conclusion of the step in question. If a step has all its premise-nodes black, then its horizontal stroke is also 'black', this being shown by a single black line. Such a step will be in $\mathcal{B}$. But if a step has at least one white premise-node, then its horizontal stroke is accordingly 'white', this being shown by two black lines with white space between them. Such a step is in $\mathcal{C}$, but not in $\mathcal{B}$.



Let $\mathcal{P}$ now be a finite set of nodes not in $\mathcal{B}$. The nodes in $\mathcal{P}$ are to be understood as 'newly adopted' initial beliefs. If, upon adopting all the new beliefs in $\mathcal{P}$, the agent finds that she now believes all the premises of some step in $\mathcal{C} \setminus \mathcal{B}$, then she is obliged to close her beliefs under that step, thereby possibly acquiring a further new belief (namely, the conclusion of the step in question). The eventual outcome of iterating the operation will be the universal closure discussed above. In symbols, it is the *universal closure of* $\mathcal{B}$ *within* $\mathcal{C}$ *with respect to* $\mathcal{P}$—abbreviated $[\mathcal{B}; \mathcal{P}]_{\mathcal{C}}$.

Let us illustrate the process of universal closure with the diagram above. Suppose we take $\mathcal{P}$ to be $\{b\}$. Then the universal closure $[\mathcal{B}; \mathcal{P}]_{\mathcal{C}}$ is easily seen to be the 'black' part of the following diagram:



By talking of nodes and of steps involving them, I am dealing with ideas familiar to workers in AI and computer science. What I have been calling 'belief-systems' modelled by means of nodes and steps are better known to workers in these areas as *dependency-networks*. Intuitively, a dependency-network is a model of a rational agent's belief-system, whose nodes represent the sentences or propositions believed.

Dependency-networks can in general contain *sub*networks. The task of contracting a network T with respect to one of its nodes p will be the task of finding a suitable subnetwork: one which does not contain the node p, but which satisfies the rational closure requirement with respect to the transitional steps in T, and which is, in a suitable sense to be explicated below, *maximal*. That is to say, the subnetwork R in question must be the outcome of a *minimally mutilating* process of contraction of the original network T. And the subnetwork R must itself be closed with respect to the transitional steps in T.

Why require such closure? Because the subnetwork R is supposed to represented a new rational 'reflective equilibrium' for the agent. Such an equilibrium will have been achieved by the agent after carrying out the contraction-process in question only if she avails herself of all the justificatory transitions *that she knows about*. The

justificatory transitions in question are exactly the transitional steps in T. We must remember that in 'giving up' a step, the agent is not losing, or eschewing, or no longer claiming to be aware of, the *justificatory transition* that it represents; rather, she is simply reaching a state in which she no longer believes the conclusion of that step, and no longer believes all its premises. The step itself, however, is still there for her to use; and use it she ought, when rationally closing her new system of beliefs, which is a *subsystem* of her original system T. Put graphically: in contractions, what can change is not the overall structure of the diagram, but only the colors of certain of its nodes (from black to white, or *vice versa*).

I shall explicate the notion of maximality below. It is important to realize that in general a network T might contain more than one maximal closed subnetwork not containing p. That is, the operation of contracting T with respect to p can in general yield more than one result. This will be reflected in the non-deterministic character of any algorithms that one might devise for performing contractions.

I have encountered the complaint, from theorists in the AGM-tradition, that any 'procedure for contraction' should tell one *exactly which* contraction to adopt. But this is not, in general, possible! Contraction is not *functional*. That is, contraction is not an operation with uniquely defined values. No matter what extra information one avails oneself of, in the way of relative entrenchment of beliefs (say), there is always the logical and mathematical possibility that *there is more than one equally good contraction of a given system of beliefs with respect to any chosen one of the beliefs therein*. My account has the virtue of accepting this possibility (which tends to be the rule rather than the exception), and turning to the task of how one accounts for *all* the eligible contractions in the multiple-valued case. Indeed, the contraction algorithm(s) to be described in another paper systematically generate all the eligible contractions; and in the implementation that I favor, in Prolog, this is achieved by back-tracking from one good solution to another, until all the equally good solutions have been enumerated.

5 The modelling: formal treatment

5.1 Nodes and steps

Definition 5.1 A pair (U, x) is called a *step* just in case
(1) U is a non-empty finite set of nodes and x is a node, and
(2) x is in U if and only if $U = \{x\}$.
U is called the *premise set* of the step (U, x); x is called the *conclusion* of the step (U, x).

Some terminological clarification is in order here. What I am here calling steps are called 'justifications' in the JTMS-literature. I prefer the shorter, neutral designation because not every step is a justification in the usual epistemological sense. In general, justifications (in the usual sense) would consist of more than one step. Note also that 'premises', in the JTMS-literature, are taken to be assumptions that cannot be retracted. This too is a non-standard use of the logician's word 'premise'. The logician speaks of premises for a step of inference, without necessarily endorsing those premises as true.

If U=$\{y_1, \ldots, y_n\}$ then the step (U, x) can be written symbolically as

$y_1, \ldots, y_n \,|\, x$.

It is easy to see that for a given x there are only two possible forms of step with x as conclusion:
$y_1, \ldots, y_n \,|\, x$ where each y_i is distinct from x $(1 \leq i \leq n)$; and
$x \,|\, x$.
I shall call these *transitional* and *initial* steps respectively.

5.2 Universal closure of belief-systems

The informal explanation of universal closure given in the previous section should enable the reader to understand the following statement, in pseudocode, of the algorithm for universal closure.

Algorithm 5.1: UNIVERSALCLOSURE($\mathcal{C}, \mathcal{B}, \mathcal{P}$)

Steps $\leftarrow \mathcal{C}$
Closure $\leftarrow \mathcal{B} \cup$ the set of initial steps involving members of $\mathcal{P}$
while some step in *Steps* but not in *Closure* has each of its premises standing as the conclusion of a step in *Closure*
do {
  NewSteps $\leftarrow$ the set of all such steps
  Closure $\leftarrow$ *Closure* $\cup$ *NewSteps*
  Steps $\leftarrow$ *Steps* $\setminus$ *NewSteps*
}
return (*Closure*)

The algorithm must terminate, since $\mathcal{C}$ is finite. I say that it determines the *universal* closure (of $\mathcal{B}$ within $\mathcal{C}$ with respect to $\mathcal{P}$) because of the word 'each' in the **while** condition.

Universal closure is exceptionally simple. Indeed, even the prima-facie more difficult problem of deciding whether a given step can be derived using finitely many given steps is solvable in linear time. In this connection, see Dowling and Gallier [6]. The problem is exactly that of querying a Prolog program with finitely many *propositional* clauses. A conditional Prolog clause such as '*a* :- *b*, *c*.' is precisely the transitional step $b, c \,|\, a$. An unconditional Prolog clause '*a*.' is precisely the initial step $a \,|\, a$. To find whether, say, the step $p, q, r \mid s$ follows from finitely many other steps, simply form the Prolog program consisting of the clauses corresponding to the latter steps, adjoin the Prolog clauses '*p*.', '*q*.' and '*r*.' to the program, and query the expanded program with *s*.

5.3 Dependency-networks

Definition 5.2 Let T be a set of steps, called *T-steps*. For any step (U, x) in T, U is said to be *x-generating* or *x-justifying* (in T).

Definition 5.3 Again, let T be a set of steps. T is a *dependency-network* if and only if T is finite and

(1) for every (U, x) in T and for every y in U, there is some (V, y) in T; and
(2) if (U, x) and (W, x) are in T, then W is not a proper subset of U.

Condition (1) in the definition of a dependency-network ensures that every node involved in any way in T is justified in T. Any conclusion x of a T-step (U, x) is justified simply because of the premise set U of that step.

Condition (2) is the rational constraint discussed earlier. It ensures that a dependency-network only ever uses the most succinct justifications available. There is no point in having the step (U, x) in a dependency-network if for some proper subset W of U we have (W, x).

I shall write '$x\,|\,x$' more economically as $|\,x$.

Remark 1: It is evident from the definition that a dependency-network need not have a well-founded justificatory structure. Note, however, that the non-well-foundedness is confined to looping, and does not arise from infinite descending chains. This is because networks are finite.

Remark 2: In general there can be more than one T-step of the form (U, x) for any given node x. Condition (2) in the definition of a dependency-network tells us that each such U is a 'minimal generating set' for x: that is, U implies x but (as far as the agent is aware) no proper subset of U implies x. In future I shall often suppress the adjective 'minimal'; but it should be borne in mind that it always applies.

Definition 5.4 $|\,T\,|$, the *core* of T, is the set of nodes x such that for some U, T contains the step (U, x).

An agent whose beliefs are modelled by a dependency-network T believes exactly what is in $|\,T\,|$, and for the reasons articulated by the steps in T. An initial node in T will be a node p for which T contains the initial step $(\{p\}, p)$.

Definition 5.5 A node q is said to be 'in' a dependency-network T (and conversely: T 'contains' q) just in case some step (U, q) with conclusion q is a member of T. (*Cf.* Doyle [7], p. 234.) Recall that a dependency-network is defined as a set of steps. Note that every member of any such premise set U will also be 'in' T, because of condition (1) in the definition of a dependency-network. Thus any node involved in any T-step is 'in' T. With this understanding of the preposition 'in', we can say (loosely) that a node is in $(T \setminus R)$ when what is meant, strictly, is that it is in $(|\,T\,| \setminus |\,R\,|)$.

Definition 5.6 R is a *subnetwork* of T just in case every step of R is a step of T and R is a dependency-network. I also say that the network T *extends* the subnetwork R.

Remark 3: Note that a subnetwork of T, since it is a dependency-network in its own right, is 'closed upwards' modulo T. That is to say, a subnetwork of T will not have any nodes of T as initial nodes if they are not also initial in T. Instead, the subnetwork will supply for each of its non-initial nodes at least one of the patterns of

justification with which it is furnished within the containing network T.

Example to illustrate Remark 3. Let T be the dependency-network

$$|p,\ \ |q,\ \ p|r,\ \ q|r.$$

The node r is the only non-initial node. Any subnetwork of T that contains r will have to contain also one of its patterns of justification within T. As it happens, there are two distinct such patterns: one leading from the initial node p, the other leading from the initial node q. Thus the only possible proper, non-empty subnetworks of T are

$$|p,\ \ p|r \qquad \text{and} \qquad |q,\ \ q|r.$$

Both these subnetworks terminate their justifications for r on initial nodes. When closing upwards modulo T, however, one does not in general always terminate in this fashion on initial nodes. For example, consider the non-well-founded network

$$p|q,\ \ q|p,\ \ p|r,\ \ q|r.$$

The only subnetwork of T that contains the node r is T itself. For, as we close upwards from r modulo T, we must include either p or q as a node. Each of these, in turn, requires the other, in order to have any T-justification at all. But neither of them is initial.

Remark 4. Not every subset of a dependency-network is a subnetwork.

Example to illustrate Remark 4. $\{(\{b\}, a)\}$ is a subset but not a subnetwork of the dependency-network $\{(\{b\}, a), (\{a\}, b)\}$. $\{(\{b\}, a)\}$ fails to be a dependency-network because b enjoys no justification within it.

Now let R be an arbitrary subnetwork of T.

Definition 5.7 R is *closed downwards* modulo T if and only if for every step (U, x) in T, if $U \subseteq |R|$, then (U, x) is in R.

Definition 5.8 The *T-closure* of R is the least subnetwork S of T satisfying the following:
(1) R is a subnetwork of S; and
(2) S is closed downwards modulo T.

This definition has as an immediate consequence that the T-closure of R is closed upwards modulo T. (See Remark 3 above.)

I shall write $[R]_T$ for the T-closure of R, and sometimes suppress mention of T when T is clear from the context. An intuitive understanding of T-closure is as follows. The subnetwork R of T, before such closure, will, as a dependency-network in its own right, contain justifications for every one of its nodes (as a conclusion). But it might also contain all the premises (*i.e.* members of U) of some T-step (U, x) without containing the conclusion x itself. The requirement of T-closure is that R should then be expanded by adding (U, x) to R and thereby justifying x 'within' R.

It does not follow that every conclusion of T itself would in due course be added in this way. The closure process could be carried out to its fullest possible extent while yet the T-closure fall properly short of T itself. This would happen when $|R|$ failed to encompass enough of the members of $|T|$ for the T-steps to effect a full enough expansion during closure. Clearly also, a proper subnetwork R of T can be properly contained in its own T-closure. The following remark summarizes these observations.

Remark 5. In general we have that R is a (possibly proper) subset of its T-closure $[R]_T$, which in turn is a (possibly proper) subset of T.

Example to illustrate Remark 5. $R = \{(\{a\}, a), (\{b\}, b)\}$ is a proper subnetwork of
$T = \{(\{a\}, a), (\{b\}, b), (\{a, b\}, c), (\{d\}, d)\}$
but the T-closure of R is $\{(\{a\}, a), (\{b\}, b), (\{a, b\}, c)\}$. So here we have R properly contained in its own T-closure, which in turn is a proper subnetwork of T.

Remark 6: $[R]_T$ can be computed from R and T in polynomial time.

Definition 5.9 A subnetwork R of T is *T-closed* if and only if R is its own T-closure, that is, just in case $R = [R]_T$.

5.4 Contraction formally defined

I shall impose three basic adequacy requirements on any contraction R of a dependency-network T with respect to any of its nodes p.

Definition 5.10 The following conditions on a subnetwork R of T are individually necessary and jointly sufficient for R to be a *contraction of T with respect to p* :

1. (Honesty) R is a T-closed subnetwork of T.
2. (Success) $|R|$ does not contain p.
3. (Minimal Mutilation) R is inclusion-maximal with regard to (1) and (2).

Remember that T-closure involves only the application of steps that are never called into question during the contraction process.

The Honesty condition says that one should not be able to 'close' $T{-}p$ any further by means of T-steps. Thus a contraction contains all nodes known to be consequences of any nodes that it contains.

The Success condition says that $T-p$ should not contain any step of the form (U, p). Put another way, p should not be in $|T{-}p|{=}|[T{-}p]_T|$.

The Minimal Mutilation condition ensures that the contraction is as economical as possible, in the sense that one gives up as little as possible in the way of T-steps when passing from T to $T-p$. The condition seeks to conserve the outputs of past computational effort in developing T. As de Kleer [5], p. 129, put it: 'Thus inferences, once made, need not be repeated ...'. I am therefore requiring the contraction to be a maximally non-(p generating), T-closed subnetwork of T. So a contraction $T{-}p$ will contain all those T-steps compatible with the requirement that p not be in $|T{-}p|$.

6 Results

6.1 The complexity of contraction

First I show that the contraction problem (understood as a decision problem) is in NP. To be precise, the decision problem in question is this:

> Given T and given an integer k, decide whether there is a T-closed subnetwork R not containing p, which is of size at least k.

Theorem 6.1 The contraction problem is in NP.

PROOF. The NP algorithm guesses R of size k and decides, in polynomial time, whether it is T-closed and is a subnetwork and does not contain p. ∎

Theorem 6.2 The contraction problem is NP-complete.

PROOF. By Theorem 6.1, the contraction problem is in NP. It remains to show that some NP-complete problem can be transformed in polynomial time into the contraction problem. To this end, consider the class of networks all of whose transitional steps have exactly two self-justifying premises, and conclusion p. Clearly any contraction of such a network (with respect to p) that is of size k will be determined by a non-p-generating subset, of size k, of the set $\mathcal{P}$ of self-justifying premises. Thus the problem of finding a contraction of size k with respect to p boils down to the problem of finding a subset of $\text{card}(\mathcal{P}) - k$, whose members are to be removed from the network. (The length of input for the latter problem is no less than some fixed fraction of the length of input for the contraction problem; such a linear factor will not affect the resulting complexity class for the contraction problem.) But this is isomorphic to the NP-complete Vertex Cover (VC) Problem, which is to find a subset (of a given size) of the set of vertices of a graph that contains, for each edge in the graph, at least one of the two vertices incident on that edge. (See Garey and Johnson [10], p. 46.) In the case of contraction I have described, we are looking, analogously, for a subset (of a given size) of the set of premises involved in two-premise steps of the network that contains, for each two-premise step in the network, at least one of its premises. (Naturally, we seek to minimize this subset in order to maximize the resulting contraction.) The isomorphism with graphs is easy to see. Construe nodes in the network as vertices in a graph. The (unordered) pair-sets of premises of the steps can then be construed as edges of the graph, which are (unordered) pair-sets of vertices. One's choice of a set of premise-nodes to be removed (at least one from each pair-set) in order for the common conclusion of all the two-premise steps to have no justification then corresponds to a choice of a set of vertices in the graph that covers each edge, by containing at least one vertex from each edge, i.e. from each pair-set of vertices. ∎

A good contraction algorithm would aim to enumerate all the contractions of a given network T with respect to any of its nodes p, without any particular interest in maximum-*sized* contractions. A contraction, by definition, is inclusion-maximal in a certain regard, and there will in general be many a contraction that is not of maximum size. The method of proof of Theorem 6.2, however, shows that the problem of searching for a contraction is NP-hard.

7 Discussion

7.1 Significance of our results for extensions of JTMS-theory

As far as *giving up* beliefs is concerned, extant JTMS-theory provides only the algorithmic operation of 'retracting an enabled assumption'. In the terminology of this paper, this is the restricted operation of contracting with respect to an *initial* node. (See Forbus and de Kleer [9], ch. 7.) This restricted operation is deterministic, and it is easy to provide for it a polynomial-time algorithm.

By contrast, I have shown here that the general, non-deterministic operation of contracting with respect to *any* node, whether initial or non-initial, is intractable. My contraction operation includes the JTMS-theorist's operation of assumption-retraction as a special case, and generalizes it appropriately for the retraction of nodes in general, in what is obviously the canonical way. Retracting a node p (equivalently: contracting with respect to a node p) is a general operation of obvious importance. The upshot of our investigations is therefore: JTMS-theory cannot be fully general in its treatment of important operations on dependency-networks except at the cost of intractability.

7.2 Logical problems are usually intractable

In any area of logic, it would be naive to expect our algorithms (when we have them) to be tractable, in the technical sense of being computable in polynomial time. After all, the simplest kind of logical problem—that of determining, of a given sentence of a propositional language, whether it is has a satisfying truth-value-assignment—is the original and most famous NP-complete problem (Cook [4]). So the problem of deciding theoremhood in classical propositional logic is co-NP-complete. Changing 'classical' to 'intuitionistic' makes matters prima facie worse: theoremhood in intuitionistic propositional logic is P-space complete (Statman [15]). In well-known systems of *relevant* propositional logic, the decision problem can be even worse. Indeed, theoremhood in the Anderson-Belnap system R of relevant logic (Anderson and Belnap [1]) is not even decidable (Urquhart [20]). Its well-known decidable fragment LR (Thistlethwaite *et al.* [19]) obtained by dropping the distributivity axiom, is at best ESPACE-hard, at worst space-hard in a function that is primitive recursive in the generalised Ackerman exponential (Urquhart, [20]).

Adding the first-order quantifiers 'some' and 'all' to the logical system brings undecidability almost everywhere. Classical first-order logic is decidable when only monadic predicates are involved, but is undecidable as soon as one adds a single two-place predicate (Church [3], [2]). Intuitionistic first-order logic is even worse: it becomes undecidable as soon as there are two one-place predicates. (This is a result of Saul Kripke. See Dummett [8], pp. 209-213.)

One should not be at all surprised or dejected, then, to find that the problem of belief-contraction is 'intractable' in the sense of being NP-complete or possibly worse. The interesting challenge was how to conceive of the problem in a manner sufficiently amenable to complexity analysis in the first place. It is in this spirit that we have conducted the foregoing investigations. The result has been encouraging. We have discovered that theory-contraction, conceived in this way, is of the lowest complexity that one could have expected it to be. Moreover, theorists of contraction who object

(misguidedly, in my view) to representing belief-systems in my finitary fashion now have to contend with the prospect that their own 'contraction decision-problems' (in so far as they might be able to define any, on their approaches) have to contend with NP-completeness as a *lower* bound of computational complexity. If NP-completeness is bad news for the dependency-network theorist, then it will be just as bad, if not worse, for *AGM*-theorists—if ever they produce any computational implementations of their contraction functions.

7.3 Future work

In planned sequels to this paper, which are already written, I specify various successively less greedy versions of a contraction algorithm; provide Prolog programs that implement them; and demonstrate how these programs produce intuitively satisfying results on all extant contraction problems canvassed in the literature.

There are also two main ways in which the simple modelling set out above could be extended, so as to take into account other features of belief-systems considered important by the epistemologist. The first extension would enable one to take into account, when performing contractions, a relation of *relative entrenchment* among nodes. This would be a partial ordering in general, providing information of the form 'if either a or b has to be given up, then it should be a rather than b'.

The second extension of the simple modelling would enable one to represent the agent as willing and able, in the course of a contraction, to give up one of the *steps* within the dependency-network. The present modelling treats the steps themselves as immune to revision, and allows only nodes to be given up. Allowing steps themselves to be vulnerable in contractions, however, can have dramatic knock-on effects, especially when one considers that a given step might be an instance of a more general rule which would itself be impugned should the step be given up. A fully detailed study of this problem is well under way, but its detailed results, both computational and philosophical, will have to await another occasion. Suffice it to say here that the modelling in this paper can be extended so as to allow for the retraction of steps as well as nodes. This is a possibility which, once explored, reveals why thinkers are rightly loathe to tamper with their logic—for the knock-on effects are enormous. It is also a possibility whose closer scrutiny commends an inviolable core of logic, namely the logic needed in order to underpin the contraction-process itself. Finally, as far as retracting steps as well as nodes is concerned: for the theoretical purposes of the present paper, it has been important *not* to countenance this possibility. That the contraction problem is NP-complete when *only nodes* are given up is all the more arresting.

In closing, I should point out that I have opted for a considerable degree of abstraction. I have abstracted away from details such as the internal logical structure of the beliefs represented by nodes, and what underlying logic might be employed to justify steps within a belief system. I have sought instead to provide a representation of what is essentially involved in the process of contraction. It remains to be investigated whether indeed my chosen level of theoretical representation might have some 'psychological reality'. Perhaps it is (or could be) part of the design of a good cognitive system that its 'contraction and revision module' operate at the level of abstraction employed here. The linguistic system furnishes complex grammatical structures for

the expression of one's beliefs; and one's logical system generates various (derived) steps, for the deductive development of one's beliefs. For the purposes of orderly, rational and efficient revision or updating, however, the only structure that is required is that captured by the kind of dependency-networks employed above. Between the actual sentences of one's language, and the nodes in such networks, there need only be a system of 'pointers'. Contraction and revision can proceed at the more abstract level of nodes and steps, prescinding altogether from distracting questions about the internal logical structure of the nodes themselves, and the kind of logic that justifies the steps.

References

[1] Alan Ross Anderson and Nuel D. Belnap, Jr. *Entailment: The Logic of Relevance and Necessity. Vol. I.* Princeton University Press, 1975.

[2] Alonzo Church. Correction. *Journal of Symbolic Logic*, 1:101–102, 1936.

[3] Alonzo Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1936.

[4] S. A. Cook. The complexity of theorem-proving procedures. *Proc. 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[5] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[6] W. F. Dowling and J. H. Gallier. Linear-time alrgorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1:267–284, 1984.

[7] John Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[8] Michael Dummett. *Elements of Intuitionism.* Clarendon Press, Oxford, 1977.

[9] K. Forbus and J. de Kleer. *Building Problem-Solvers.* MIT Press, Cambridge, MA, 1993.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Bell Laboratories, Murray Hill, NJ, 1979.

[11] Peter Gärdenfors. *Knowledge in Flux.* MIT Press, Cambridge, MA, 1988.

[12] S. O. Hansson. Belief contraction without recovery. *Studia Logica*, 50:251–260, 1991.

[13] Isaac Levi. *The Fixation of Belief and Its Undoing.* Cambridge University Press, 1991.

[14] R. Niederée. Multiple contraction. a further case against Gärdenfors' Principle of Recovery. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change. Lecture Notes in Artifical Intelligence 465.* Springer Verlag, Berlin, Heidelberg, New York, 1987.

[15] Richard Statman. Intuitionistic propositional logic is polynomial space complete. *Theoretical Computer Science*, 9:67–72, 1979.

[16] Neil Tennant. Changing the theory of theory change: Towards a computational approach. *British Journal for Philosophy of Science*, 45:865–897, 1994.

[17] Neil Tennant. Changing the theory of theory-change: reply to my critics. *British Journal for Philosophy of Science*, 48:569–586, 1997.

[18] Neil Tennant. On having bad contractions: or, no room for recovery. *Journal of Applied Non-Classical Logics*, 7:241–266, 1997.

[19] Paul Thistlewaite, Michael A. MacRobbie, and Robert K. Meyer. *Automated Theorem-Proving in Non-Classical Logic. Research Notes in Theoretical Computer Science.* Pitman, London and Wiley, New York, 1987.

[20] Alasdair Urquhart. The undecidability of entailment and relevant implication. *Journal of Symbolic Logic*, 49:1059–1073, 1984.

Received September 2003