APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN ENGINEERING PROBLEMS

Proceedings of the 1st International Conference,
Southampton University, U.K.
April 1986

Volume II

Editors:

D. Sriram
R. Adey

A Computational Mechanics Publication

Springer-Verlag
Berlin Heidelberg New York Tokyo

D SRIRAM

Department of Civil Engineering
M.I.T.
Cambridge
Massachusetts 02139
USA

R ADEY

Computational Mechanics Inc
Suite 6200
400 West Cummings Park
Woburn MA 01801
USA

British Library Cataloguing in Publication Data

International Conference on the Application of Artificial Intelligence in Engineering Problems (1986: University of Southampton) Proceedings of the 1st International Conference on the Application of Artificial Intelligence in Engineering Problems, Southampton University U.K., April 1986.
1. Engineering design—Data processing
2. Artificial intelligence 3. Pattern recognition systems
I. Title II. Sriram, D. III. Adey, R.
620'.00425'0285 TA174

ISBN 0-905451-47-3

ISBN 0-905451-47-3 Computational Mechanics Centre, Southampton
ISBN 3-540-16349-2 Springer-Verlag Berlin Heidelberg New York Tokyo
ISBN 0-387-16349-2 Springer-Verlag New York Heidelberg Berlin Tokyo

Hypothetical Reasoning

Joao P. Martins
Departamento de Engenharia Mecanica, Instituto Superior Tecnico, Av. Rovisco Pais, 1000 Lisboa, Portugal.
Stuart C. Shapiro
Department of Computer Science
State University of New York at Buffalo, 226 Bell Hall, Buffalo, N.Y. 14260, U.S.A.

ABSTRACT

We present a domain-independent reasoning system that is able to perform hypothetical reasoning. A person working with our system may raise hypotheses, request the system to reason from them, adding the results obtained to a knowledge base, and may discard any of the hypotheses raised, which automatically makes inacessible to the reasoning program every piece of knowledge depending on the hypothesis (or hypotheses) discarded.

The novelty of our approach to hypothetical reasoning lies in the way that we switch reasoning contexts. The system is able to return to a previous state of reasoning, without performing any backtracking at all. With our approach the system maintains several contexts, defined by different and even competing sets of hypotheses, may switch back and forth between contexts, and may compare results obtained in different contexts.

INTRODUCTION

In this paper we discuss hypothetical reasoning, one of the features of a large system, the MBR System [Martins 83], [Martins and Shapiro 83], [Martins and Shapiro 84]. MBR was developed with the goal of being able to detect and recover from the contradictions generated in a reasoning system (that is, MBR is a belief revision system). There are two characteristics in MBR that distinguish it from most other belief revision systems (for example, [Doyle 78, 79, 83], [Goodwin 82, 84], [McAllester 78, 80], [McDermott 83] and [Thompson 79]): (1) it is based on a logic developed to support belief revision systems; (2) it relies on the manipulation of assumptions, not justifications (an excellent discussion of this second aspect can be found in [de Kleer 84]).

The paper very briefly presents the logic underlying MBR (the SWM system), present a discussion on how a computer program interprets the propositions of the logic and then discusses how MBR performs hypothetical reasoning.

Belief revision and hypothetical reasoning are important in engineering applications (see, for example, [Raulefs 84]). In the last part of the paper we show an example of hypothetical reasoning using MBR.

THEORETICAL FOUNDATIONS - THE SWM SYSTEM

The SWM system (after Shapiro, Wand and Martins) is the logic underlying MBR. In this section we present SWM. When discussing a logic, there are two aspects to consider, its syntax and its semantics.

The syntax of a logic includes a set of formation rules and a set of rules of inference. The set of formation rules determines which formulas are legal in the logic. These formulas are called well-formed formulas, wffs for short. We will assume standard formation rules for wffs with "~", "v", "&", "->" as connectives, and ∀ and E as quantifiers. See, for example, [Lemmon 78, pp.44 and 104]. The set of rules of inference (the deductive system) specifies which conclusions may be inferred from which premises. Given an argument (P,c) (a premisse-conclusion argument is an ordered pair (P,c) in which P is a set of propositions, called premisses and c is a single proposition, called conclusion) we say that c is deducible from P, written P |- c, if there is a sequence of rules of inference which when applied to P produces c.

The semantics of a logic concerns the study of the conditions under which sentences are true or false. The semantics are completely determined by the specification of two things, the interpretations of the language (every possible assignment of a particular object to each particular member of the language) and the truth conditions for it (what it means for a given sentence to have a given truth value in a given interpretation). We say that the argument (P,c) is valid if there is no interpretation in which each sentence in P is true and in which c is false. If (P,c) is valid, we write P |= c.

There is nothing about validity in the deductive system, and there is nothing about deducibility in the semantics. Although syntax and semantics are separate parts of a logical system, and thus deducibility and validity are intensionally distinct, they must fit together properly in order for the system to make any sense. A logic is said to be sound if and only if every argument deducible in its deductive system is valid according to its semantics. A logic is said to be complete if and only if every argument valid according to its semantics is deducible in its deductive system. Given a "reasonable" semantics, a logic can be unsound due to "wrong" rules of inference; and a logic can be incomplete due to the lack of necessary rules of inference or due to rules of inference that are too constraining. The SWM system is an incomplete logic, since several arguments valid according to its semantics are not deducible in its deduction system. This fact should not be regarded as a drawback of the logic but rather as a feature that makes it attractive for its intended applications.

The first step towards formally analyzing arguments consists of providing precise meaning for everyday terms like "and", "or", "if", "if...then...", "every", "some", etc. In the process of translating an informal argument into a formal one, some of the features of the informal argument are lost. The important point is to keep in the model those features that are of interest to the modeler. Therefore, when assigning meaning to the logical terms, one should bear in mind which features of the informal arguments one wants to preserve in their formal counterparts.

In our case, SWM was developed to support belief revision systems, the main goal is to keep a record of propositional dependencies. Keeping propositional dependencies is very important in belief revision systems: when something goes wrong (a contradiction is detected) we should be able to find out what propositions lead to the contradiction, i.e. what are the propositions on which the contradiction depends on. Our our approach adopts the meaning of the logical connectives used in classical logic and builds a deductive system that blocks some unwanted deductions (resulting in an incomplete system). Most of the blocked deductions involve the introduction of irrelevancies.

Let us informally discuss what types of information we need in our logic (a detailed discussion of these issues can be found in [Martins 83] and [Martins and Shapiro 84]). One of the fundamental problems that any logic underlying a belief revision system has to address is how to keep track of and propagate propositional dependencies. This is important, because, in the event of detection of a contradiction, one should be able to identify exactly which assumptions were used in the derivation of the contradictory propositions. We don't want to blame some assumption irrelevant to the occurrence of the contradiction as the culprit for the contradiction, and, when looking for the possible culprits for a contradiction, we don't want to leave out any assumption possibly responsible for the contradiction. In the field of logic, the relevance logicians also want to keep track of what propositions were used to derive any given proposition. Relevance logicians have developed mechanisms to keep track of propositional dependencies and to prevent the introduction of irrelevancies. One way of doing this (used in the FR system of [Anderson and Belnap 75, pp.346-348] and in the system of [Shapiro and Wand 76]) consists of associating each wff with a set, called the origin set, which references every hypothesis (non-derived proposition) used its derivation. The rules of inference are stated so that all the wffs derived using a particular hypothesis will reference this hypothesis in their origin sets. Whenever a rule of inference is applied, the origin set of the resulting wff is computed from the origin sets of the parent wffs. In order to guarantee that the origin set only contains the hypotheses actually used in the derivation of the wff, some of the applications of the rules of inference allowed in classical logic are blocked. Most of this mechanism was adopted in the SWM system.

Besides the dependency-propagation mechanism, there is another advantage in using relevance logic, to support belief revision systems. In classical logic a contradiction implies anything; thus, in a belief revision system based on classical logic, whenever a contradiction is derived it should be discarded immediately. In a relevance-logic-based belief revision system, we may allow the existence of a contradiction in the knowledge base without the danger of filling the knowledge base with unwanted deductions. In a relevance logic-based belief revision system all a contradiction indicates is that any inference depending on any hypothesis underlying a contradiction is of no value. In this type of systems we can even perform reasoning in a knowledge base which is known to be inconsistent.

Another important issue in belief revision systems, reflected in our logic, consists in the recording of the conditions under which contradictions may occur. This is important because once we discover that a given set is inconsistent, (a set is inconsistent if a contradiction may be derived from it. A set is consistent just in case it is not inconsistent). We may not want to consider it again, and even if we do want to consider it, we want to keep in mind that we are dealing with an inconsistent set. In the SWM system, contradictions are recorded by associating each wff with a set, called the restriction set, that contains information about which sets unioned with the wff's origin set produce an inconsistent set. When new wffs are derived, their restriction sets are computed directly from the restriction sets of the parent wffs, and when contradictions are detected all the wffs whose origin set references any of the contradictory hypotheses has its restriction set updated to record the newly discovered contradictory set. Similarly to what happens with origin sets, SWM makes sure that restriction sets don't have any more information than what they should.

In addition, for the proper application of some rules of inference, it is important to know whether a given wff was introduced as a hypothesis or was derived from other wffs. In order to do this, we associate each wff with an identifier, called the origin tag that tells whether the wff is a hypothesis, a normally derived proposition, or a special proposition, that if treated regularly, would introduce irrelevancies into the knowledge base.

Formally, the SWM system deals with objects called supported wffs. A supported wff consists of a wff and an associated triple containing an origin tag (OT), an origin set (OS), and a restriction set (RS). The set of all supported wffs is called the knowledge base. We write A | t,a,r to denote that A is a wff with OT "t", OS "a", and RS "r", and we define the functions ot(A)=t, os(A)=a and rs(A)=r.

The OS is a set of hypotheses. The OS of a supported wff contains those (and only those) hypotheses that were actually used in its derivation.

The OTs range over the set {hyp, der, ext}: hyp identifies hypotheses, der identifies normally derived wffs within SWM, and ext identifies special wffs whose OS was extended.

An RS is a set of sets of wffs. A wff, say A, whose RS is {R1,...,Rn} means that the hypotheses in os(A) added to any of the sets R1,...,Rn produces an inconsistent set. The RS of an extended wff contains every set which unioned with the wff's OS will produce a set that is known to be inconsistent. It is important to distinguish between a set being inconsistent and a set being known to be inconsistent. An inconsistent set is one from which a contradiction can be derived; a set known to be inconsistent is an inconsistent set from which a contradiction has been derived. The goal of adding RSs is to avoid re-considering known inconsistent sets of hypotheses. Our rules of inference guarantees that the information contained in the RS is carried over to the new wffs whenever a new proposition is derived.

The rules of inference of SWM guarantee that (see [Martins 83]):

1. The OS of a supported wff contains every hypothesis that was used in its derivation.

2. The OS of a supported wff only contains the hypotheses that were used in its derivation.

3. The RS of a supported wff records every set known to be inconsistent with the wff's OS.

4. The application of rules of inference is blocked if the resulting wff would have an OS known to be inconsistent.

The OT and OS of a proposition reflect the way the proposition was derived: the OS contains the hypotheses underlying that proposition, and the OT represents the relation between the proposition and its OS. The RS of a proposition reflects our current knowledge about how the hypotheses underlying that proposition relate to the other hypotheses in the knowledge base. Once a proposition is derived, its OT and OS remain constant; however, its RS changes as the knowledge about all the propositions in the knowledge base does. We are not addressing here the problem of multiple derivation of the same proposition. A discussion of this aspect of SWM can be found in [Martins 83] and [Martins and Shapiro 84].

A CONTEXTUAL INTERPRETATION FOR SWM

We now discuss how a program using SWM should interpret SWM's wffs. In this section, we provide what we call a contextual interpretation for SWM. We use the word "contextual interpretation" instead of just "interpretation" for the following two reasons: On the one hand, we want to stress that we are not providing an interpretation for SWM in the logician's sense of the word; on the other hand, we want to emphasize that our definition of truth depends on the notion of context. This contextual interpretation defines the behavior of an abstract revision system (i.e., not tied to any particular implementation), which we call MBR (Multiple Belief Reasoner).

We will assume that MBR works with a knowledge base containing propositions that are associated with an OT, OS, and RS (in SWM's sense). Propositions are added to the knowledge base according to the rules of inference of SWM.

We define a context to be a set of hypotheses. A context determines a Belief Space (BS), which is the set of all the hypotheses defining the context and all the propositions that were derived exclusively from them. Within the SWM formalism, the wffs in a given BS are characterized by having an OS that is contained in the context. The set of contexts represented in the knowledge base is the power set of the set of hypotheses existing in the knowledge base.

Any operation performed within the knowledge base (query, addition, deletion, etc.) will be associated with a context. We will refer to the context under consideration, i.e., the context associated with the operation currently being performed in the knowledge base, as the current context. While the operation is being carried out, the only propositions that will be considered are the propositions in the BS defined by the current context. This BS will be called the current belief space. A proposition is said to be believed if it belongs to the current BS. We can look at contexts as delimiting smaller knowledge bases (namely, the Belief Spaces) within the knowledge base. The only propositions in the knowledge base that are retrievable are those propositions that belong to the current BS.

A common goal of belief revision systems is to stay away from contradictions, i.e., to avoid the simultaneous belief of a proposition and its negation. Taking this into account, it would seem natural to constrain contexts to be consistent sets of hypotheses, not just any sets of hypotheses.

However, it may be the case that in MBR one desires to perform reasoning within the BS defined by an inconsistent context. In SWM, the existence of contradictions is not as damaging as in classical logic, in which anything can be derived from a contradiction. Thus, in MBR one may not want to bother discarding hypotheses after a contradiction is detected, since the contradiction will not affect the entire system. For these reasons, in MBR, the condition that a context is not known to be inconsistent will not be compulsory but rather advisable if one doesn't explicitly want to perform reasoning in a BS that is known to be inconsistent. The reason why it is advisable is that within a BS defined by a context not known to be inconsistent some simplification can be considered during the application of the rules of inference (for details refer to [Martins 83]).

HYPOTHETICAL REASONING

Hypothetical reasoning is reasoning made from one or more hypotheses whose truth value is unknown, doubtful or even known to be false. Based on these three possibilities, Rescher defines three different kinds of hypothetical reasoning:

> "hypothetical inference is reasoning which derives a conclusion from premises one or more of which is problematic (of unknown truth-status) or belief-contravening (negating some accepted belief and thus taken to be false) or outright counterfactual (i.e., actually known to be false)" [Rescher 64, p.1] (italics in the original)

We now consider the possible applications of these kinds of reasoning and how MBR handles them.

Problematic Reasoning is reasoning in which one (or more) of the premises have an unknown truth value, being possibly true or possibly false with no definite view being held. Such type of reasoning is useful in contingency-planning, i.e., look ahead reasoning with hypotheses that may or may not happen, so that the reasoner can be prepared to handle possible future situations.

Belief-contravening reasoning is reasoning done from premisses which "conflict with accepted beliefs upon grounds that are inductive or probabilistic rather than logico-deductive" [Rescher 64, p.4]. The premises are "believed" to be false although there are no logical grounds yet to show their falsehood. This is the kind of reasoning used in the well known reductio ad absurdum proof method.

Counterfactual reasoning is reasoning made from premises which are known to be false. This is the kind of reasoning usually known as "reasoning for the sake of argument".

MBR deals in a similar way with the different kinds of hypothetical reasoning: when an hypothesis which originates hypothetical reasoning is introduced it is treated as a regular hypothesis and MBR proceeds without worrying about it. All the wffs derived from this hypothesis include in their OS a reference to it.

In the case of contingency planning and reasoning for the sake of argument MBR does whatever inferences are needed and when the hypothesis which originated hypothetical reasoning is dropped from the current context, all the wffs that were derived from it will be ignored by the system. The word ignored is the most appropriate since those wffs will neither be erased nor marked as disbelieved, (as is the case in other systems, for example [Doyle 79]) they just will not be considered for the application of further rules of inference. If later on, that hypothesis is raised again all the wffs which were derived from it are immediately available for further deductions.

In the case of a proof by reductio ad absurdum, when a contradiction is found and MBR is faced with the task of negating one of the hypothesis it will negate the hypothesis which originated the belief-contravening reasoning using SWM's rules that deal with contradictions (the statement of these rules can be found in [Martins 83], [Martins and Shapiro 83], and [Martins and Shapiro 84]).

PROOF OF ENTAILMENTS

In this section we present an example of application of problematic reasoning: proof of entailments. In a natural deduction based system, (see for example, [Fitch 52]) to prove that A->B one initiates a subproof with hypothesis A and tries to deduce B within that subproof. When trying to deduce B in this subproof one can use wffs which belong to the subproofs containing the subproof initiated by the hypothesis A.

SWM is a natural deduction-based system which does not rely on an explicit structure of subproofs (see [Martins 83]). However, subproofs can be simulated in MBR by means of contexts and BSs and a such feature is demonstrated in this section. When MBR is asked to prove the entailment A->B, it creates a new context by adding the hypothesis A to the current context (the context in which the entailment was asked to be proved) and tries to prove B within the BS defined by this new context. The original context represents the set of all hypotheses initiating the subproofs containing the subproof where the hypothesis A is raised. The BS defined by that context contains all the propositions in those subproofs.

The reason why this falls under problematic reasoning (in Rescher's sense) is that MBR does not have to know the truth value of A prior to its tentative assumption to prove that A->B, and therefore it is reasoning from one premise (A) which is possibly true or possibly false.

The example that we show in this section was obtained using an implementation of MBR (the SNeBR system). In SNeBR, propositions are represented in the SNePS network [Shapiro 79]. The example presented uses SNePSLOG [McKay and Martins 81] a logic interface to SNePS: when using SNePSLOG one enters propositions in predicate logic notation, SNePSLOG translates them into SNePS network nodes, performs the actions requested and then translates the results generated back into predicate logic notation. The numbers associated with the wffs represent the number of the SNePS nde which represents the wff.

We will use the following example is from [McCawley 81, p.25]: suppose that a murder has been committed and that we know that "whoever committed the murder left by the window" and that "anyone who left by the window would have mud on his shoes". These propositions are represented in the knowledge base, respectively, by wffs, wff17 and wff21 (Figure 1).

```
* build ∀(x) [murderer(x) -> left-by(x, window)]
  wff 17 built

* build ∀(x) [left-by(x, window) -> has-mud(x, shoes)]
  wff 21 built
```

Figure 1
Deduction rules

Suppose that MBR is asked to show that if the butler committed the murder then he has mud in his shoes (Figure 2). This proof entails raising the hypothesis that the butler committed the murder (Figure 2) and performing deduction based on such hypothesis. Notice that MBR does not know whether or not the butler committed the murder, but just wants to know what will follow if such was the case. The reasoning followed by MBR is shown in Figures 2 and 3.

```
* deduce(murderer(butler) -> has-mud(butler, shoes))
  context (wff17 wff21)

I wonder if
murderer(butler) -> has-mud(butler, shoes)
holds within the BS defined by the context (wff17 wff21)

Let me assume that
wff22: murderer(butler)
```

Figure 2
Run (part 1, creating hypothesis)

It should be noticed that after raising the hypothesis that the butler is the murderer (wff22, Figure 2), the context considered by MBR is {wff17, wff21, wff21} instead of {wff17, wff21} which is the context in which the request was made.

It should also be pointed out that after the interaction is concluded, wff22 bear no relationship with the wffs in the BS defined by the context {wff17, wff21}. However, wff22 is built in the network. If, after this interaction we would ask whether has-mud(butler, shoes) in a BS containing the context {wff17, wff21, wff22} the answer would be "yes" without any deduction being performed.


```
I wonder if
  has-mud(butler, shoes)
holds within the BS defined by the context (wff22 wff17 wff21)

Let me try to use the rule
  left-by(butler, window) -> has-mud(butler, shoes)

I wonder if
  left-by(butler, window)
holds within the BS defined by the context (wff22 wff17 wff21)

Let me try to use the rule
  murderer(butler) -> left-by(butler, window)

I wonder if
  murderer(butler)
holds within the BS defined by the context (wff22 wff17 wff21)

I know
  murderer(butler)

since
  murderer(butler)
I infer
  left-by(butler, window)

since
  left-by(butler, window)
I infer
  has-mud(butler, shoes)

since
  has-mud(butler, shoes)
was derived under the assumption
  murderer(butler)
I infer
  murderer(butler) -> has-mud(butler, shoes)
```

Figure 3
Run (part 2, deducing new information)

CONCLUDING REMARKS

We discussed MBR, a belief revision system, briefly described some of the concepts of the logic that underlines MBR; discussed how MBR performs hypothetical reasoning; and showed an example. The example presented was obtained from an actual run just by slightly changing the syntax of the propositions.

MBR is implemented in SNePS, a powerful knowledge representation system. A distinguishing characteristic of MBR is that it is based on a logic (SWM) designed with the goal of supporting belief revision systems. SWM associates each proposition with all the hypotheses used in its derivation and with all the hypotheses with which it is known to be incompatible The SWM formalism guarantees that (1) The origin set of a supported wff contains every proposition that was used in its derivation. (2) The origin set of a supported wff only contains the hypotheses that were used in its derivation. (3) The restriction set of a supported wff records every set known to be inconsistent with the wff's origin set. (4) The application of the rules of inference is blocked if the resulting wff would have an origin set known to be inconsistent.

The queries to MBR are associated with a context, the network retrieval function only considers the propositions in the BS defined by that context. When a contradiction is detected, after selecting one hypothesis (or several hypotheses) as the culprit for the contradiction, the removal from the network of all the propositions depending on such hypothesis (hypotheses) is done just by dropping it (them) from the context being considered. Afterwards these propositions will no longer be in the BS under consideration and thus will not be considered by MBR. This approach is extremelly useful for performing hypothetical reasoning.

ACKNOWLEDGEMENTS

Many thanks to John Corcoran, Jon Doyle, Donald McKay, Ernesto Morgado, J. Terry Nutter, William J. Rapaport and members of the SNePS Research Group for their criticisms and suggestions.

This work was partially supported by the National Science Foundation under Grant MCS80-06314 and by the Instituto Nacional de Investigacao Cientifica (Portugal), under Grant No. 20536; Preparation of this paper was supported in part by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB DC 20332 under contract No. F30602-85-C-0008.

REFERENCES

1. Anderson A. and Belnap N., Entailment: The Logic of Relevance and Necessity Vol.1, Princeton University Press, 1975.

2. de Kleer J., "Choices without Backtracking", Proc. AAAI-84, pp. 79-85.
3. Doyle J., "Truth Maintenance Systems for Problem Solving", Technical Report AI-TR-419, Massachussets Institute of Technology, AI Lab., 1978
4. Doyle J., "A Truth Maintenance System", Artificial Intelligence, Vol.12 No.3, pp.231-272, 1979.
5. Doyle J., "The Ins and Outs of Reason Maintenance", Proc. IJCAI-83, pp.349-351.
6. Fitch F., Symbolic Logic: An Introduction, Ronald Press, 1952.
7. Goodwin J. W., "A Improved Algorithm for Non-Monotonic Dependency Net Update", Research Report LiTH-MAT-R-82-23, Software Systems Research Center, Linkoping Institute of Technology, Sweden, August 1982.
8. Goodwin J.W., "WATSON: A Dependency Directed Inference System", Proc. Non-Monotonic Reasoning Workshop, pp.103-114, 1984.
9. Lemmon E.J., Beginning Logic, Hackett Publishing Company, 178.
10. Martins J., "Reasoning in Multiple Belief Spaces", Ph.D. Dissertation, Technical Report 203, Department of Computer Science, State University of New York at Buffalo, 1983.
11. Martins J. and Shapiro S., "Reasoning in Multiple Belief Spaces", Proc. IJCAI-83, pp.370-373.
12. Martins J. and Shapiro S., "A Model for Belief Revision", Proc. Non-Monotonic Reasoning Workshop, pp.241-294, American Association for Artificial Intelligence, 1984.
13. McAllester D., "A Three-valued Truth Maintenance System", Technical Report Memo 473, Massacussets Institute of Technology, AI Lab., 1978.
14. McAllester D., "An Outlook on Truth-Maintenance", AI Memo 551, Massachussets Institute of Technology, AI Lab., 1980.
15. McCawley J.D., Everything that Linguists have Always Wanted to know about Logic (but were ashamed to ask), The University of Chicago Press 1981.
16. McDermott D., "Data Dependencies on Inequalities", Proc. AAAI-83, pp.266-269.

17. McKay D and Martins J., "Provisional SNePSLOG User's Manual", SNeRG Technical Note 4, Department of Computer Science, State University of New York at Buffalo, 1981.

18. Raulefs Pl, "Foundation of Expert Systems for Conceptional Design in Mechanical Engineering", Memo SEKi-84-08, Fochbereich Informatik, Universitat Kaiserslautern, October 1984.

19. Rescher N., Hypothetical Reasoning, North-Holland Publishing Co., 1964.

20. Shapiro S., "The SNePS Semantic Network Processing System", in Associative Networks, Findler (ed.), pp.179-203, Academic Press, 1979.

21. Shapiro S. and Wand M., "The Relevance of Relevance", Technical Report No.46, Computer Science Department, Indiana University, 1976.

22. Thompson A., "Network Truth-Maintenance for Deduction and Modelling", Proc. IJCAI-79, pp.877-879.