

Verification of non-monotonic knowledge bases

Neli P. Zlatareva *

Department of Computer Science, Central Connecticut State University, New Britain, CT 06050-4010, USA

Abstract

Non-monotonic Knowledge-Based Systems (KBSs) must undergo quality assurance procedures for the following two reasons: (i) belief revision (if such is provided) cannot always guarantee the structural correctness of the knowledge base, and in certain cases may introduce new semantic errors in the revised theory; (ii) non-monotonic theories may have multiple extensions, and some types of functional errors which do not violate structural properties of a given extension are hard to detect without testing the overall performance of the KBS. This paper presents an extension of the distributed verification method, which is meant to reveal structural and functional anomalies in non-monotonic KBSs. Two classes of anomalies are considered: (i) structural anomalies which manifest themselves within a given extension (such as logical inconsistencies, structural incompleteness, and intractabilities caused by circular rule chains), and (ii) functional anomalies related to the overall performance of the KBS (such as the existence of complementary rules and some types of rule subsumptions). The corresponding verification tests are presented and illustrated on an extended example. © 1997 Elsevier Science B.V.

Keywords: Verification; Performance evaluation; Testing; Validation; Quality assurance; Anomaly detection; Non-monotonic theories; Knowledge-Based Systems; Expert systems; Truth maintenance systems; Default reasoning

1. Introduction

In the recent years, a significant effort has been made to tackle different aspects of Knowledge-Based Systems (KBS) quality assurance. Although success has been achieved in certain areas, especially in the development of formal methods for KBS verification and building of verification tools, most of this work is limited to rule-based systems employing monotonic inference. More advanced KBSs (often referred to as a *second generation* KBSs) are non-monotonic. To cope with incomplete domain specifications, these systems employ some form of a closed world assumption (for example, negation in logic programming and defaults in frame representation systems

and semantic networks), which may lead to the replacement of already inferred conclusions with new ones if additional evidence becomes available later in the inference process. As a result, the consistency of the knowledge base (KB) can be violated. To handle such cases, some non-monotonic systems incorporate a belief revision facility. There is no guarantee, however, that belief revision will necessarily succeed in recovering the consistency of the KB. Moreover, new semantic errors can be introduced as a result of the belief revision process [17]. This is why we argue that non-monotonic KBSs must undergo quality assurance procedures to ensure the structural correctness of their knowledge bases and their problem-solving adequacy.

So far, little attention has been paid in the V & V community to quality assurance of non-monotonic

* Corresponding author.

KBSs. This is partially due to the fact that these systems are still at the stage of research prototypes, and partially due to the widely spread perception that the nature of non-monotonic reasoning suggests incompleteness and inconsistency in the domain specification, and thus functional problems are unavoidable. We believe that quality assurance procedures can help in detecting and correcting a large number of potential functional errors.

The first attempt to address quality assurance needs of non-monotonic KBSs was undertaken within the EVA project [2]. This research, however, was mostly concerned with defining *what* a non-monotonic KBS is, rather than with identification and characterization of classes of anomalies that a non-monotonic KBS may contain. The types of errors examined were redundancies, inconsistencies, and incompleteness. It was suggested that the detection of such errors can be carried out in the same way as in monotonic KBS, namely by describing higher-order constructs (metapredicates) representing different types of errors. Metapredicates are entered as queries to the KBS, and the generated answers reveal all of the errors of a particular type detected in the KB-theory.

There are at least two problems with this approach:

- It is computationally very expensive, which makes it inapplicable for large-scale KBSs.
- It deals with a limited set of anomalies that can be defined as metapredicates.

Similar ideas have recently been presented in Ref. [1], where an attempt was made to adopt methods designed for verification of monotonic rule-based systems to non-monotonic systems.

In this article, we advocate the idea that the nature of non-monotonic KBSs leads to a different expression of common types of errors such as inconsistencies, circularities and redundancies, and suggests new types of errors which have no counterpart in monotonic KBSs. To the best of our knowledge, no serious study has been made to identify and characterize errors unique to non-monotonic KBSs. In an attempt to fill this gap, in Section 2 we summarize and adapt for the purposes of KBS verification some results on error taxonomy from machine learning literature, and expand this taxonomy with new types of anomalies which are likely to alter structural or

functional properties of a non-monotonic KBS. In Section 3, a brief introduction to non-monotonic systems based on default logic and truth maintenance systems is given, and an example of a non-monotonic theory is presented. Sections 4 and 5 introduce and illustrate the proposed verification technique. In Section 6, we summarize the results of this research and state our plans for future work.

2. Error taxonomy in non-monotonic KBs

In Mitchell et al. [10], the following three types of structural problems in non-monotonic theories are outlined:

1. The *incompleteness* problem, which results in the fact that some conclusions cannot be inferred because relevant information is missing.
2. The *inconsistency* problem, which results in generating logical contradictions due to wrong assumptions.
3. The *intractability* problem, which results in computationally prohibitive deductions.

In Ref. [11], incompleteness errors are divided into:

- Errors due to missing knowledge.
 - Errors due to the lack of sufficient detail in relevant knowledge.
- Similarly, inconsistency errors are divided into:
- Errors caused by wrong knowledge.
 - Errors caused by wrong assumptions.

Errors due to missing knowledge can be further subdivided into two categories depending on the error indicator:

- Errors revealed by the existence of *irrelevant rules*, i.e., rules which do not contribute to the inference of any final hypothesis.
- Errors revealed by the existence of *unreachable hypotheses*.

Errors due to the lack of sufficient detail in relevant knowledge can be subdivided into:

- Errors revealed by the existence of *irrelevant facts*, i.e., facts upon which no final hypothesis depends.
- Errors revealed by logical and/or semantic contradictions detected during verification.

As an instance of the intractability problem, we consider circularities among inference rules. In some

cases, circular rules may result in infinite recursions. We also show that circular rule chains in a non-monotonic theory may cause logical and/or semantic contradictions.

Another type of a functional anomaly in non-monotonic KBSs can be caused by redundant rules if these violate the predictability requirement. According to the predictability requirement, each of the correctly solved test cases should be handled properly by the system in the future; that is, the system should never generate different solutions for the same test case because of potential conflicts among multiple rules that may fire at the same time.

A different class of anomalies unique to non-monotonic systems are those related to the existence of multiple extensions of a non-monotonic theory. We consider two types of anomalies in this group:

- Anomalies caused by redundant rules, which are generalizations of existing default rules.
- Anomalies caused by the existence of complementary rules, i.e., rules forcing the same conclusion under contradictory assumptions or data.

Anomalies in this class are hard to detect, because they typically do not violate the structural properties of the knowledge base. However, as it is shown in Section 5.7, such rules may violate the functional correctness of the KBS, and therefore they must be identified and reviewed during the verification process.

3. Non-monotonic KB-theories employing default rules

In this article we consider non-monotonic knowledge bases incorporating default rules. The presented verification framework, however, can be applied to knowledge bases employing circumscription [9] (computational difficulties associated with this formalism limit its application in real-world KBSs), or to logic programs (although these can be handled by conventional verification techniques such as DIVER [18]).

Default logic [12] is one of the most popular formalisms for representing non-monotonic theories. It can be implemented by a non-monotonic Truth Maintenance System (TMS) [4] or with some approximation by an Assumption-Based TMS [3]. At-

tempts to use different types of TMSs in commercial environments for building KBSs started in the mid-eighties [7], and today many second generation KBSs and advanced KBS development tools (KEE [6], for example) incorporate some kind of a TMS to handle non-monotonic specifications or to improve the efficiency of the inference process.

Default logic employs two types of inference rules:

- Deductive (or monotonic) rules, whose conclusions are logically true beliefs. These have the form $\frac{A}{C}$, where A is a monotonic prerequisite for conclusion C .
- Default (or non-monotonic) rules, which have the form $\frac{A:C}{C}$. Here conclusion C holds only under the assumption that it is consistent to believe in C .

Introduction of default rules in a KB-theory creates a possibility for logical inconsistencies. One form of a logical inconsistency is manifested in the existence of multiple extensions of a KB-theory. To handle this problem, a good deal of work has been done in finding a way of selecting among different extensions. Most of this work is based on the notion of *more specific* knowledge [5,15], or employs the idea presented in Ref. [13] to explicitly introduce priorities between competing rules by using *semi-normal* defaults. The latter are rules of the form $\frac{A:C \wedge B_1 \wedge \dots \wedge B_n}{C}$, where $C \wedge B_1 \wedge \dots \wedge B_n$ is the justification for holding C . Justifications of this form are useful for checking the rule's consistency. However, once inferred the rule's conclusion is no longer related to its justification. This may result in generation of semantically wrong conclusions. For example, given $\text{Adult}(\text{Tom}), \text{Student}(\text{Tom}) \vee \text{Priest}(\text{Tom})$, and the two semi-normal defaults

$$\frac{\text{Adult}(x):\text{Married}(x) \wedge \neg \text{Student}(x)}{\text{Married}(x)},$$

$$\frac{\text{Adult}(x):\text{Married}(x) \wedge \neg \text{Priest}(x)}{\text{Married}(x)},$$

one can infer that Tom is married, although neither students nor priests are married according to the theory.

To a certain extent, this problem is fixed by TMSs where justifications are explicitly stored as a

part of belief specification. In this context, default rules are represented as follows:

- Normal defaults: $(I_1, I_2, \dots, I_k)(\neg C) \rightarrow C$, where $C: (I_1, I_2, \dots, I_k)(\neg C)$.
- Semi-normal defaults: $(I_1, I_2, \dots, I_k)(O_1, O_2, \dots, O_n) \rightarrow C$, where $C: (I_1, I_2, \dots, I_k)(O_1, O_2, \dots, O_n)$.

Here I_1, I_2, \dots, I_k are the monotonic supporters for conclusion C , called the ‘IN-list’ of the justification; and $O_1, O_2, \dots, O_n, \neg C$ are the non-monotonic supporters for C , called the ‘OUT-list’ of the justification. In both cases, conclusion C holds if all of the rule’s monotonic supporters hold, and none of its non-monotonic supporters hold.

An example of a non-monotonic theory represented in a TMS-format is given below (this example is extracted from the knowledge base, which was kindly placed by Torsten Schaub at my disposal for the purposes of this research). This example is used throughout this article to illustrate the proposed verification framework.

Deductive rules

$R_1: (\text{kid}(x))() \rightarrow \text{has_toys}(x)$

$R_3: (\text{professor}(x))() \rightarrow \text{academic}(x)$

$R_4: (\text{priest}(x))() \rightarrow \text{academic}(x)$

Non-monotonic rules: normal defaults

$R_5: (\text{student}(x))(\neg \text{adult}(x)) \rightarrow \text{adult}(x)$

$R_6: (\text{student}(x))(\text{employed}(x)) \rightarrow \neg \text{employed}(x)$

$R_7: (\text{student}(x))(\text{married}(x)) \rightarrow \neg \text{married}(x)$

$R_9: (\text{academic}(x))(\neg \text{adult}(x)) \rightarrow \text{adult}(x)$

Non-monotonic rules: semi-normal defaults

$R_2: (\text{adult}(x))(\text{professor}(x)) \rightarrow \neg \text{has_toys}(x)$

$R_8: (\text{adult}(x))(\text{student}(x)) \rightarrow \text{employed}(x)$

$R_{10}: (\text{adult}(x))(\text{student}(x)) \rightarrow \text{married}(x)$

$R_{11}: (\text{academic}(x))(\text{professor}(x)) \rightarrow \neg \text{employed}(x)$

$R_{12}: (\text{academic}(x))(\text{priest}(x)) \rightarrow \text{has_toys}(x)$.

Input data: $\text{professor}(x)$, $\text{student}(x)$, $\text{kid}(x)$, $\text{priest}(x)$.

Final hypotheses: $\text{married}(x)$, $\text{employed}(x)$, $\text{has_toys}(x)$, and their negations.

Semantic constraint: $\text{adult}(x) \wedge \text{kid}(x)$.

4. Distributed verification revised

The distributed verification method, which has originally been developed for verification of monotonic rule-based KBSs [16], follows the so-called *operationalization* approach to KBS verification

[8,14]. The underlying idea is to obtain an operational version of the KB-theory by using an alternative inference engine which is known to be correct and complete. During the operationalization process all anomalies hidden in the original KB-theory become explicit and thus easily detectable. To compute the operational version of the KB-theory, the distributed verification method employs the Contradiction-tolerant TMS (CTMS) [17]. The operationalization process consists of the following three steps: (i) conversion of the KB-theory into a CTMS-theory; (ii) generation of the stable extension of the latter, where all contradictions that the KB-theory implicitly contains are revealed and represented as formulas of a special type; and (iii) generation of the operational version of the KB-theory by computing the transitive closure of the formulas comprising the stable extension.

Similar to a non-monotonic TMS, the CTMS has two types of rules called *T*-rules and *U*-rules. The former are monotonic rules, while the latter behave non-monotonically as a result of the rule revision performed by the CTMS if a logical contradiction is encountered. Although the semantics of *U*-rules is different from the semantics of default rules, *U*-rules can be easily adjusted to represent both normal and semi-normal defaults. *U*-rules have the form $(T_1, \dots, T_n)(U_1, \dots, U_m) \rightarrow A$, where T_1, \dots, T_n are monotonic supporters for A representing the minimum evidence required to consider A as a plausible belief, while U_1, \dots, U_m are additional arguments for A which, if become true, will increase the plausibility of A . Thus, each *U*-rule can be viewed as a scheme of plausible rules each instance of which accounts for a subset of *U*-arguments found to be true. Specific instances are represented by the so-called ‘duplicate rules’ which, in turn, can be:

- *T*-duplicates. These have the form $(T_1, \dots, T_n, U_1, \dots, U_m)() \rightarrow A$, where $A': (T_1, \dots, T_n, U_1, \dots, U_m)()$.
- *U*-duplicates. These have the form $(T_1, \dots, T_n, U_{i_1}, \dots, U_{i_k})(\{U_1, \dots, U_m\} \setminus \{U_{i_1}, \dots, U_{i_k}\}) \rightarrow A$, where $A'': (T_1, \dots, T_n, U_{i_1}, \dots, U_{i_k})(\{U_1, \dots, U_m\} \setminus \{U_{i_1}, \dots, U_{i_k}\})$

There is one *U*-duplicate per each *U*-argument of the original rule.

Given a non-monotonic KB-theory $\langle SR, SF, SH, SC \rangle$, where *SR* is a set of monotonic and non-monotonic rules, *SF* is a set of input data (facts), *SH* is a

set of final hypotheses, and SC is a set of semantic constraints, its conversion into a CTMS-theory $\langle DB, R \rangle$ is performed as follows.

(1) Each fact $F_i \in SF$ is represented as a formula $F_i^f:() \in DB$, each hypothesis $H_j \in SH$ is represented as a formula $H_j^h:() \in DB$, and each rule $R_i \in SR$ is represented as a formula $R_i^r:() \in DB$.

(2) Depending on the type of a KB rule, three cases of rule conversion are considered:

(a) Monotonic KB rules are translated into CTMS T -rules $(T_1, \dots, T_n, R_i)() \rightarrow H_i$, where T_1, \dots, T_n are the premises of the KB-rule, and R_i is the rule itself.

(b) Normal defaults are translated into CTMS U -rules $(T_1, \dots, T_n, R_i)(H_i) \rightarrow H_i$, where conclusion H_i is the only U -argument.

(c) Semi-normal defaults are translated into CTMS U -rules $(T_1, \dots, T_m, R_i)(U_{i_1}, U_{i_2}, \dots, U_{i_n}) \rightarrow H_i$, where $\neg U_{i_1}, \dots, \neg U_{i_n}$ are the non-monotonic arguments comprising the ‘OUT-list’ of the KB rule.

A complete set of duplicate rules must be generated for each U -rule:

- For normal defaults:
 $(T_1, \dots, T_m, H_i, R_i)() \rightarrow H_i$.
- For semi-normal defaults:
 $(T_1, \dots, T_m, U_{i_1}, R_i)(U_{i_2}, \dots, U_{i_n}) \rightarrow H_i$,
 $(T_1, \dots, T_m, U_{i_1}, U_{i_2}, R_i)(U_{i_3}, \dots, U_{i_n}) \rightarrow H_i$,
 \dots
 $(T_1, \dots, T_m, U_{i_1}, \dots, U_{i_n}, R_i)() \rightarrow H_i$.

(3) Each semantic constraint $SC_i \in SC$ is represented as a formula $C_{SC_i}^u(X_1, \dots, X_n)(C_{SC_i})$, which is not an element of the initial DB , but is added to it only if all of its premises, X_1, \dots, X_n , hold.

Provided that the KB-theory can be converted into a CTMS-theory by means of the described procedure, the first step of the verification process consists in generating the operational version of the KB-theory. The main assumption behind the operationalization approach is that all of the hidden in the KB-theory anomalies can be revealed if the entire set of input data is enabled simultaneously. In non-monotonic theories, this assumption may explicate some logical contradictions that can be successfully handled by the belief revision facility (typically implemented by a dependency-directed backtracking procedure). It is easy to see that belief revision is capable of handling contradictions, where the culprit is a datum serving as a non-monotonic supporter in a

semi-normal default. These contradictions can be removed from consideration if the input data set is divided into maximal input sets (MISs) consistent with respect to the data serving as non-monotonic supporters in semi-normal defaults.

To illustrate this step, consider the example theory introduced in Section 3. There are three input data serving as non-monotonic supporters in semi-normal defaults: $\text{professor}(x)$, $\text{student}(x)$, and $\text{priest}(x)$. Therefore, eight MISs containing different combinations of these data must be considered. Each of these MISs originates an independent operationalization process, which is carried out in the same way as for monotonic theories. That is, the stable extension of the corresponding CTMS-theory is computed by repeatedly applying the CTMS-rules until no more rules can fire, and then the operational theory is generated by computing the transitive closure of the formulas comprising the stable extension (for details, see Ref. [16]). The resulting theory, called the ‘grounded stable extension’, consists of formulas of the following three types:

- $H_i^f:(F_1, \dots, F_i, R_1, \dots, R_j)()$.
- $H_i^u:(F_1, \dots, F_i, R_1, \dots, R_j)(\neg F_k, \dots, \neg F_m, \neg H_p, \dots, \neg H_s)$.
- $C_{F_i, F_j}^u:(F_i, F_j)(C_{F_i, F_j})$.

Here F_1, \dots, F_i is called the *data set* of the formula, $\neg F_k, \dots, \neg F_m, \neg H_p, \dots, \neg H_s$ is called the *assumption set*, and R_1, \dots, R_j is called the *rule set*. C_{F_i, F_j} represents a logical or semantic contradiction between formulas with heads F_i and F_j (as a special case, $F_i = \neg F_j$).

By analyzing individual formulas or group of formulas from the grounded stable extensions, different types of anomalies and errors in the original KB-theory can be revealed by means of the verification tests presented next.

5. Verification tests for non-monotonic KB-theories

Verification process involves a number of tests intended to detect and correct the following types of anomalies and errors in non-monotonic KB-theories.

- Logical inconsistencies caused either by wrong assumptions or wrong knowledge. All cases of logical inconsistencies are revealed by looking for formulas representing logical contradictions in each of the grounded stable extensions.

- Structural incompleteness caused by missing knowledge and/or lack of sufficient detail in the relevant knowledge. All cases of structural incompleteness are revealed by looking for final hypotheses which cannot be reached by a given MIS, or by identifying irrelevant rules and irrelevant facts (i.e., rules and facts which do not participate in any data or rule set of the formulas comprising grounded stable extensions).

- Circular rule chains causing infinite recursions or logical and/or semantic contradictions. These are detected by looking for formulas whose heads or their negations are also encountered in their T -sets. It is important to note that formulas inferred by T -duplicates of normal defaults are by definition of this type. This is why they must be recognized and treated as a special case.

- Redundant rules which may or may not constitute an error in the KB-theory depending on the type of redundancy, and whether or not alternative hypotheses are allowed. There are several cases of rule redundancies in non-monotonic theories: (i) a deductive rule is subsumed by a normal default; (ii) a deductive rule is subsumed by a semi-normal default; (iii) a normal default is subsumed by a semi-normal default; and (iv) redundancy among rules of the same type.

A special case of rule redundancy in non-monotonic theories is the existence of complementary rules. Such rules cause the same conclusion to be inferred by inconsistent MISs, which clearly indicate a performance error.

All cases of rule redundancy can be identified by looking for formulas with the same head. To detect complementary rules, this procedure must be performed over all grounded stable extensions at the same time.

Next, we introduce the verification tests in more detail, and illustrate them by means of the example theory.

5.1. Test for logical inconsistencies due to wrong assumptions

Assume that instead of R_8 , the example theory contains the following normal default: $R_8^* : (\text{adult}(x))(\neg \text{employed}(x)) \rightarrow \text{employed}(x)$.

Then, the grounded stable extension originated by the maximal input set containing $\text{professor}(x)$, $\text{student}(x)$, $\text{priest}(x)$, and $\text{kid}(x)$ contains the following logical contradiction: $C_{\text{employed}}^u : (\text{employed}(x), \neg \text{employed}(x), R_6, R_8^*)(C_{\text{employed}})$.

This contradiction can be resolved if one of the two normal defaults, R_6 or R_8^* , is converted into a semi-normal default by adding one of the monotonic supporters of the other default as its non-monotonic supporter. We can (arbitrarily) choose to modify R_8^* which becomes: $R_8 : (\text{adult}(x))(\text{student}(x)) \rightarrow \text{employed}(x)$.

That is, the detected inconsistency is resolved by specializing a normal default, R_8^* , into a semi-normal default, R_8 , thus preventing it from firing when $\text{student}(x)$ is given.

5.2. Test for logical inconsistencies due to wrong knowledge

Assume that the following rule is added to the example theory: $R_i : (\text{professor}(x))() \rightarrow \neg \text{has_toys}(x)$.

Then, grounded stable extensions originated by maximal input sets where both $\text{professor}(x)$ and $\neg \text{priest}(x)$ hold, contain the following contradiction: $C_{\text{has_toys}}^u : (\text{professor}(x), R_3, R_{12}, R_i)(\text{priest}(x), C_{\text{has_toys}})$.

The detected contradiction is caused by the inconsistency between R_3 and R_{12} , on the one hand, and R_i on the other hand. Under the ‘minimum change’ rationale, we can assume that R_i represents semantically wrong knowledge, and it must be considered for deletion from the KB-theory.

5.3. Test for structural incompleteness due to missing knowledge

Assume that R_7 is missing from the example theory. Then the grounded stable extension originated by the maximal input set containing $\neg \text{professor}(x)$, $\text{student}(x)$, $\neg \text{priest}(x)$, $\text{kid}(x)$ consists of the following formulas: $DB_G = DB \cup \{\text{has_toys}(x) : (\text{kid}(x), R_1)(), \text{adult}(x) : (\text{student}(x), R_5)(), \neg \text{employed}(x) : (\text{student}(x), R_6)(), \neg \text{has_toys}(x) : (\text{adult}(x), R_2)(), C_{\text{has_toys}}^u : (\text{kid}(x), \text{adult}(x), R_1, R_2)(C_{\text{has_toys}})\}$.

Notice that contradiction $C_{\text{has_toys}}$ is caused by the

violation of a declared semantic constraint, which is why this contradiction does not indicate a structural error in the KB-theory.

One of the final hypotheses, $\text{married}(x)$, is not contained in DB_G , neither is its negation, $\neg\text{married}(x)$. This indicates that the KB-theory is incomplete. To reveal the source of the problem, the verification procedure examines all of the rules whose conclusion is either the missing final hypothesis or its negation. The only such rule here is: $R_{10}:(\text{adult}(x))(\text{student}(x)) \rightarrow \text{married}(x)$.

Assuming that this rule is semantically correct, a conclusion can be drawn that students are typically not married. This knowledge can be represented as the following normal default: $(\text{student}(x))(\text{married}(x)) \rightarrow \neg\text{married}(x)$,

If this new rule is added to the KB-theory, then $\neg\text{married}(x)$ can be inferred given $\neg\text{professor}(x)$, $\text{student}(x)$, $\neg\text{priest}(x)$, and $\text{kid}(x)$.

5.4. Test for structural incompleteness due to the lack of sufficient detail in relevant knowledge

This type of incompleteness arises if the initial set of facts, SF , does not contain all of the data required during the inference process. If a missing datum serves as a non-monotonic supporter in a semi-normal default, then two different extensions of SF must be considered:

- An extension originated by SF with the missing datum added.
- An extension originated by SF with the negation of the missing datum added.

It is easy to see that the KB-theory will infer contradictory conclusions depending on which of the two extensions is considered. Such a contradiction, however, will automatically go away if the missing fact is added to SF . This case of semantic inconsistency can be handled by the KBS control mechanism if heuristic knowledge for ranking different extensions is available. This is why an incompleteness of this type does not indicate a performance error, but rather an anomaly (lack of required data) which can be fixed by completion of the input data set.

In our example, if input datum $\text{priest}(x)$ is missing, depending on whether $\text{priest}(x)$ or $\neg\text{priest}(x)$ is assumed, two different hypotheses, $\neg\text{has_toys}(x)$

and $\text{has_toys}(x)$, will be inferred. Adding the missing datum to the initial data set will solve the detected problem.

5.5. Test for circular rule chains

Consider the following hypothetical rules:

$R_i:(\text{adult}(x))(\text{student}(x)) \rightarrow \text{married}(x)$

$R_{i+1}:(\text{married}(x))() \rightarrow \neg\text{kid}(x)$

$R_{i+2}:(\neg\text{kid}(x))() \rightarrow \text{student}(x)$

$R_{i+3}:(\text{student}(x))() \rightarrow \neg\text{married}(x)$.

Assume that $\text{adult}(x)$ and $\text{student}(x)$ are both declared as input facts, but there is no information about whether x is a student or not. Obviously, there are two MISs in this case, which contain $\text{adult}(x)$, $\text{student}(x)$ and $\text{adult}(x)$, $\neg\text{student}(x)$, respectively. Notice that the grounded stable extension originated by the former MIS will contain $\neg\text{married}(x)^t:(\text{student}(x), R_{i+3})()$, which is the expected conclusion. The grounded stable extension originated by the other MIS will contain the formula, $\text{student}(x)^t:(\text{adult}(x), \neg\text{student}(x), R_i, R_{i+1}, R_{i+2})()$, indicating that the underlying theory is intractable, because $\text{student}(x)$ can be inferred only if $\neg\text{student}(x)$ holds, which is impossible.

Let us consider how the KB-theory behaves in this case. Assuming that $\text{student}(x)$ does not hold, R_i will fire, thus generating $\text{married}(x)$. This causes R_{i+1} to fire generating $\neg\text{kid}(x)$, which in turn causes R_{i+2} to fire generating $\text{student}(x)$. The latter invalidates the already inferred conclusion $\text{married}(x)$, which in turn invalidates $\neg\text{kid}(x)$ and $\text{student}(x)$, thus causing $\text{married}(x)$ to be inferred again. That is, R_i , R_{i+1} , and R_{i+2} form a cycle which makes the KB-theory intractable. The verification procedure recognizes such errors by looking for formulas which depend on their own negations.

Circular rule chains may also cause logical and/or semantic contradictions. Assume that the following rule is added to the example theory: $R_j:(\text{has_toys}(x))() \rightarrow \text{professor}(x)$.

Given a MIS, where $\neg\text{professor}(x)$ holds, the formula $\text{professor}(x)^t:(\text{has_toys}(x))()$ will be inferred by R_j causing a logical contradiction. On the other hand, given $\text{kid}(x)$ (which participates in all MISs), this would also constitute a semantic contradiction, if there were a semantic constraint

$\text{professor}(x) \wedge \text{kid}(x)$. Notice that the belief revision facility will not recognize this contradiction, because $\neg \text{professor}(x)$ is not explicitly specified as an input datum.

5.6. Test for redundant rules

In most cases, redundant rules do not violate structural properties of a KB-theory. Sometimes, however, they may lead to alternative development of the inference process, thus violating the predictability requirement.

The test for redundant rules is applied to the rule sets of the formulas with the same head, and it checks if their data sets are identical, or one data set is a subset of another. For example, assume that: $R_k: (\text{professor}(x))() \rightarrow \text{has_toys}(x)$ is added to the example theory. Then, the following two formulas will belong to grounded stable extensions originated by maximal input sets containing both $\text{professor}(x)$ and $\neg \text{professor}(x)$:

$\text{has_toys}(x): (\text{professor}(x), R_k)()$,
 $\text{has_toys}(x): (\text{professor}(x), R_3, R_{12})(\neg \text{professor}(x))$.

It follows from here that either R_k or R_3 and/or R_{12} are redundant. Note that two of these rules, R_3 and R_k , are deductive rules, while R_{12} is a semi-normal default. From a knowledge representation point of view, monotonic rules and normal defaults are weaker than semi-normal defaults. Therefore, it is reasonable to assume that monotonic rules and normal defaults are better candidates for deletion in case of rule redundancy than semi-normal defaults. Whether or not a redundant rule must be deleted from a KB-theory depends on a particular application.

5.7. Test for complementary rules

This test is a special case of the test for redundant rules, where all grounded stable extensions are searched for formulas with the same head simultaneously. These formulas are then searched for *complementary rules*, i.e., rules which force the same hypothesis under contradictory assumptions or data. To identify such rules, both *T*-sets and *U*-sets of the formulas with the same head are examined for monotonic and/or non-monotonic supporters with opposite signs.

Consider, for example, the following hypothetical rules:

$R_l: ()(\text{priest}(x)) \rightarrow \text{has_toys}(x)$

$R_{l+1}: (\text{priest}(x))() \rightarrow \text{adult}(x)$

$R_{l+2}: (\text{adult}(x))() \rightarrow \text{has_toys}(x)$

Given $\text{priest}(x)$, conclusion $\text{has_toys}(x)$ will be inferred by means of R_{l+1} and R_{l+2} . Given $\neg \text{priest}(x)$ the same conclusion, $\text{has_toys}(x)$, will be inferred by means of R_l . Notice that this error does not violate structural properties of the KB-theory. However, it clearly points a performance error, which can be detected only if different extensions are examined simultaneously.

6. Conclusion

In this paper, we argue that non-monotonic KBSs must undergo quality assurance procedures for the following two reasons:

1. Belief revision, if such is provided, cannot always guarantee the structural correctness of the knowledge base; moreover, in certain cases it may introduce new semantic errors in the revised theory.
2. Non-monotonic theories may have multiple extensions, and some types of functional errors which do not violate structural properties of a given extension, are hard to detect without testing the overall performance of the KBS.

We have presented an extension of the distributed verification method [16] meant to perform verification on non-monotonic knowledge bases incorporating default rules. The extended conversion procedure has been introduced to account for normal and semi-normal defaults in non-monotonic KB-theories. Verification tests presented in Section 5 are intended to detect two classes of anomalies in non-monotonic KB-theories: (i) structural anomalies which manifest themselves within a given extension (such as logical inconsistencies, structural incompleteness, and intractabilities caused by circular rule chains), and (ii) functional anomalies related to the overall performance of the KBS (such as the existence of complementary rules and some types of rule subsumptions).

The presented verification framework has been tested on relatively large example theories, and the initial results demonstrate its usefulness and effi-

ciency. It is our hope that we will be able to test it on a real-world KBS to examine its limitations and expand the existing set of verification tests with new ones. Another direction of our future work is the implementation of an automated refinement procedure to enhance not only the verification process, but the overall development of a KBS as well.

Acknowledgements

This research has been partially supported by a CSU-AAUP research grant. Many thanks to Charles Neville for his useful suggestions.

References

- [1] G. Antoniou, Verification and correctness issues for non-monotonic knowledge bases, in: *Proc. European Symposium on the Validation and Verification of Knowledge-Based Systems (Eurovav-95)*, 1995, pp. 141–153.
- [2] C.L. Chang, R.A. Stachowitz, J.B. Combs, Validation of nonmonotonic knowledge-based systems, in: A. Dollas, W.T. Tsai, N.G. Bourbakis (Eds.), *Proc. 2nd International Conference on Tools for Artificial Intelligence (TAI-90)*, 1990, pp. 776–782.
- [3] J. de Kleer, An assumption-based TMS, *Artificial Intelligence (Netherlands)* 28 (2) (1986) 127–162.
- [4] J. Doyle, A truth maintenance system, *Artificial Intelligence (Netherlands)* 12 (1979) 231–272.
- [5] D. Etherington, Reasoning with incomplete information. *Research Notes in Artificial Intelligence*, Pitman Publ., 1988.
- [6] R. Fikes, T. Kehler, The role of frame-based representation in reasoning, *Communications of the ACM* 28 (9) (1985) 904–920.
- [7] R.E. Filman, Reasoning with worlds and truth maintenance in a knowledge-based programming environment, *Communications of the ACM (US)* 31 (4) (1988) 382–401.
- [8] A. Ginsberg, Knowledge-base reduction: a new approach to checking knowledge bases for inconsistency and redundancy, in: *Proc. 7th National Conference on Artificial Intelligence (AAAI 88)*, Vol. 2, 1988, pp. 585–589.
- [9] J. McCarthy, Applications of circumscription to formalizing common-sense knowledge, *Artificial Intelligence (Netherlands)* 28 (1986) 89–116.
- [10] T.M. Mitchell, R.M. Keller, S.T. Kedar-Cabelli, Explanation-based generalisation: a unifying view, *Machine Learning (Netherlands)* 1 (1986) 47–80.
- [11] S. Rajamoney, G. DeJong, The classification, detection and handling of imperfect theory problems, in: *Proc. AAAI 1988*, 1988, pp. 205–207.
- [12] R. Reiter, A logic for default reasoning, *Artificial Intelligence (Netherlands)* 13 (1980) 81–132.
- [13] R. Reiter, G. Criscuolo, On interacting defaults, in: *Proc. International Joint Conferences on Artificial Intelligence (IJCAI'81)*, 1981, pp. 270–275.
- [14] M.-C. Rousset, On the consistency of knowledge bases: the COVADIS system. *Computational Intelligence (Canada)*, 4 (2) 1988, pp. 166–170. Also in: *ECAI 88, Proc. European Conference on AI, Munich, August 1–5, 1988*, pp. 79–84.
- [15] D. Touretzky, *The Mathematics of Inheritance*. Research Notes in Artificial Intelligence, Pitman Publ., 1986.
- [16] N. Zlatareva, Distributed verification: a new formal approach for verifying knowledge-based systems, in: J. Liebowitz (Ed.), *Expert Systems World Congress Proceedings*, Vol. 2, 1991, Pergamon Press, New York, pp. 1021–1029.
- [17] N. Zlatareva, CTMS: a general framework for plausible reasoning, *Int. J. Expert Syst.: Res. Appl.* 5 (4) (1992) 229–247.
- [18] N. Zlatareva, A. Preece, An effective logical framework for knowledge-based systems verification, *Int. J. Expert Syst.: Res. Appl.* 7 (3) (1994) 239–260.

Neli P. Zlatareva is an Associate Professor of Computer Science at Central Connecticut State University in New Britain, Connecticut. She received her MS in Computer Science with the highest honor from the Higher Institute of Transport Engineering in Moscow, Russia, and her PhD in Computer Science from the Higher Institute of Mechanical and Electrical Engineering in Sofia, Bulgaria. She also holds a Habilitation Degree in Artificial Intelligence from the Bulgarian Academy of Sciences. Zlatareva's research interests include quality assurance of Knowledge-Based Systems, knowledge base refinement and incremental learning, non-monotonic reasoning and truth maintenance systems. She is an author/co-author of more than 60 publications in these areas.