Influence of anomalous truth values on logical inference in $V^{TF}$-logics as a basis for verification of rule-based systems

L V Arshinskiy$^1$, V L Arshinskiy$^2$, M P Dunaev$^2$, M S Nitezhuk$^1$

$^1$ Irkutsk State Transport University, 15, Chernyshevskogo str., Irkutsk, 664074, Russia
$^2$ Irkutsk National Research Technical University, 83, Lermontova str., Irkutsk, 664074, Russia

E-mail: larsh@mail.ru

Abstract. The paper is devoted to the problem of expert systems knowledge bases verification. The methodological basis of verification is logics with vector semantics in the $V^{TF}$-logics form. The knowledge model is a rule-based system. The issues of algorithmization of contradictions and other problems detection are considered in the paper. Algorithmization is based on the characteristic features of in $V^{TF}$-logics inference. It is shown that in the formalism under consideration, the anomalous truth of a small premise generates the same conclusion (a large premise is considered strictly true). Anomalies such as falsity, uncertainty, and contradiction are considered. The problems of reducing the computational complexity of algorithms are considered.

1. Introduction

Despite well-known criticism, expert systems (ES) are often important components of managing for complex technical, industrial, and other systems (see, for example, [1-4]). This is mainly due to the weakly structured and poorly formalized complex subject areas (SA). Knowledge models used in ES are often an alternative approach to modeling. The advantage of knowledge models, compared, for example, with neural networks, which are also used in this area [5-8], is an explicit and editable representation of knowledge. In artificial neural networks, knowledge is "buried" in the network architecture and its changing is possible only through retraining, or conceptual rethinking of the network. In this regard, modeling by means of knowledge will be in demand at least in the nearest future. In this meaning, knowledge models are also of interest when managing critical infrastructures, where timely response to a situation and its meaningful understanding is an important element of safety.

Like any other model, the knowledge model may contain errors and must be verified. A large number of studies have been devoted to verification of knowledge models (see, for example, [9-19]). The main emphasis is placed on rule-based models as the most popular in practice [14-17, 19]. Many authors consider such classes of anomalies of rule-based knowledge bases (KB) as a violation of consistency (inconsistency, cyclicity, redundancy, intersections) and a violation of integrity (incompleteness in a narrow sense, lack of links, incorrect attribute values) [14]. Static and dynamic anomalies are also considered [15, 16]. The first ones are those that are identified by analyzing the KB structure. The others are anomalies that appear during the inference engine operation. Static anomalies...
include, for example, incompleteness, redundancy, and the presence of facts that are not used anywhere. The dynamic anomalies include inconsistency of the knowledge system. Thus, incompleteness may be manifested by the presence of unattainable, missed, terminating rules, and redundancy – by duplicate rules and rules that do not lead to any goal [16]. The interest in anomalies of rule-based models is linked with that fact; they are easily modified unlike network models. Moreover, we can modify the KB without coordinating the next rule with the existing ones. This external ease is the reason for the discussed problems (modularity is also typical of logical models, but the situation there is similar, so we are talking only about rules).

Despite the relatively large number of possible KB artefacts, the contradiction in KB is the most interesting case for researchers. This is, for instance, [17]:

- contradiction within the rules when parts of the rules exclude each other (for example, \( a \rightarrow \neg a \), here \( \rightarrow \) and \( \neg \), respectively, implication and negation);
- contradictions between rules that lead to mutually exclusive conclusions, such as \( a \) and \( \neg a \);
- internal inconsistency of reasoning chains, when facts that contradict known facts are introduced into the system;
- contradictory chain of inference when the same data set creates a mutually exclusive situation.

In addition, the contradiction may not be known immediately. For example, if it is hidden in synonymous words or SA features [16, 18, 19].

Static anomalies are a simple case. They can be identified at the stage of creating the knowledge base. Dynamic ones require computational verification by starting the solver. Authors would like to:

- automatize this procedure;
- reduce the number of solver launches to identify artefacts.

There are other anomalies also covered in this paper.

2. Logics with vector semantics as an artefacts detecting tool

To solve this problem, we will use the capabilities of logics with vector semantics in the form of VTF-logics [20-23]. Their distinctive feature is the ability to perform inference in conditions of data deficit and/or inconsistency. The classic inference is impossible in this situation. At the same time, we consider only artefacts of the type of uncertainty and/or contradictions in the formal sense without reference to the content of the SA. Errors in the form of logical circles, non-use facts in rules, and other static anomalies can be detected without using the logical inference procedure.

In VTF-logics, uncertainty are formalized by the truth vector \( \langle 0;0 \rangle \), a contradiction by any \( ||a|| = (a';a) \), where \( a',a \in [0,1] \). Here \( a' \) is a measure of confidence that the proposition \( a \) is true, and \( a' \) that \( a \) is false. In classical inference, uncertainty in this sense occurs if there is no inference step that results in the truth of some proposition. In other words, \( a \) does not participate in the inference or it is terminated. The conclusion may fail if the antecedent of the rule is always false. The problem arises in case of conflict when one branch of the inference is conclusion \( a \), and the other is \( \neg a \).

Application of the VTF-logics allows you to bring the inference to the end. If there is uncertainty or contradiction at any of the stages, they «stretch» to the end. Moreover, in certain cases, we can stretch the false premise influence to the end of inference. It generates a false result at the output. Detection of such a situation is a signal of the presence of an artifact.

The core of the product is the implication \( i = a \rightarrow b \). The inference rule is «modus ponens» in the form:

\[
a, a \rightarrow b \mid b; ||b|| = ||a \& i|| = ||\neg a \vee i||.
\]

After the colon, the method of calculation of the truthfulness of the conclusion is specified. It is based on truth of small and large premise: \( a \) and \( i = a \rightarrow b \). The «&» is a symbol of conjunction, «\( \vee \)» is the symbol of disjunction; \( \leftrightarrow \) is the range of values of the vector of the truth. In this case it is the rectangle in the square \([0,1]\times[0,1] \). It is defined by its upper left and lower right corners: \( ||a \& i|| \) and \( ||\neg a \vee i|| \). The value of \( ||b|| \) in this case is equal to:
\[ \|b\| = \langle [a^* \bullet i^*, a^* \oplus i^*]; [a^* \bullet i^*, a^* \oplus i^*] \rangle. \]  

(2)

Here \( \bullet \) is a triangular norm, \( \oplus \) is a triangular co-norm with an additional axiom:

\[ x \bullet y = 1 - (1-x) \oplus (1-y); \]

or

\[ x \oplus y = 1 - (1-x) \bullet (1-y). \]

Several situations are considered:

(i). One of the premises is false or strictly false: \( \langle 0; 1 \rangle \). For a strictly false small premise, the area of possible values of the conclusion truth vector is the square \([0,1] \times [0,1]\). It makes the value \( \|b\| \) as blurry and inaccurate as possible. In this sense, the «entropy of truth» becomes maximal.

With a strictly false large premise, the value of the truth of the conclusion falls into the area close to the strict lie:

\[ \|b\| = \langle [a^* \bullet 0, a^* \oplus 0]; [a^* \bullet 1, a^* \oplus 1] \rangle = \langle [0, a^*]; [a^* \oplus 1, 1] \rangle. \]

This allows us to conclude that the falsity of a small premise makes the truth of the conclusion unknown; the falsity of a large one generates a lie in the conclusion.

(ii). One of the premises is uncertain: \( \langle 0; 0 \rangle \). For a small premise it turns out:

\[ \|b\| = \langle [0 \bullet i^*, 0 \oplus i^*]; [0 \bullet i^*, 0 \oplus i^*] \rangle = \langle [0, i^*]; [0, i^*] \rangle. \]

The uncertainty of a small premise placed the truth vector of the conclusion (somewhere) in the region \([0, i^*] \times [0, i^*]\). The borders of this area are set by the vector \( \|b\| \).

If the implication is uncertainty, we get:

\[ \|b\| = \langle [a^* \bullet 0, a^* \oplus 0]; [a^* \bullet 0, a^* \oplus 0] \rangle = \langle [0, a^*]; [0, a^*] \rangle. \]

In other words, the area of truth of the vector \( \|b\| \) is a square with sides \([0,a^*]\)×\([0,a^*]\).

If the small premise is strictly true or close to it, the uncertainty of the large premise generates the uncertainty of the conclusion (the vector \( \|b\| \) is located near the point \( \langle 0; 0 \rangle \)). If it is strictly false, we come to the already known square \([0,1] \times [0,1]\).

Thus, the uncertainty of both small and large premises leads to uncertainty in the conclusion. At the same time, with a high degree of likelihood of a small premise, the uncertainty generated by a large premise is significant.

(iii). One of the premises is contradictory: \( \langle 1; 1 \rangle \). If it is a small premise then:

\[ \|b\| = \langle [1 \bullet i^*, 1 \oplus i^*]; [1 \bullet i^*, 1 \oplus i^*] \rangle = \langle [i^*, 1]; [i^*, 1] \rangle. \]

is the area \([i^*, 1] \times [i^*, 1]\) – relatively close to \( \langle 1; 1 \rangle \) (more precisely, to \([1,1] \times [0,1]\)). If a large premise is inconsistent, that:

\[ \|b\| = \langle [a^* \bullet 1]; [a^* \bullet 1] \rangle. \]

That is, with a sufficiently reliable small premise, the region close to \( \langle 1; 1 \rangle \) is obtained again. If it is strictly false or close to it – the entire or almost the entire square \([0,1] \times [0,1]\).

The inconsistency of both small and large premises leads to a contradictory conclusion.

The points (i)-(iii) allow automatizing the search for anomalies. Automatizing looks like a special logical inference at the end of which an artifact appears. The actual location of the artifact is detected by a back tracing of the reasoning chain.

3. Algorithmization of artefact detection

We assume that the knowledge model artefacts are a consequence of the content of the KB. The logical inference shows up the artefacts. Moreover, artefact exists if there is a combination of input data from a valid set that leads to undesirable consequences. These include:
- non-participation of a knowledge unit in the inference for all valid truth values of input assumptions;
- uncertainty of the conclusion for some of the valid values truth of the input facts and the presence of not used anywhere facts;
- inconsistency of the conclusion for some of the valid values truth of the input assumptions.

Verification makes it possible to establish that such situations do not occur.

"Silent" rules (the first case) are detected by counting the number of calls to each rule for all valid input facts. If there were no calls, the product may be excluded from the KB.

The second case is related to the break in the chain of reasoning. For example, if antecedent is false. This leads to the uncertainty of the consequent, which will affect the terminal facts (hypotheses).

Or, if you do not break the chain by the condition of insufficient truth of the small premise, to the maximum entropy of the hypothesis's truth: [0,1]\times[0,1].

Finally, a contradiction occurs when one chain produces a statement and the other a denial of the same fact. Because of evidence combining (it is a necessary element of plausible inference) according to the rule:

\[ \|b_1 \lor b_2\| = \langle b_1^+ \oplus b_2^+; b_1^- \oplus b_2^- \rangle, \]

(3)

where \( \|b_i\| = \langle b_i^+; b_i^- \rangle \), \( \|b_2\| = \langle b_2^+; b_2^- \rangle \) we also get a contradiction in the output. Here \( \lor_2 \) is the second form of disjunction [20-22] calculated by the formula:

\[ \|a \lor_2 b\| = \langle a^+ \oplus b^+; a^- \oplus b^- \rangle. \]

If \( \|b\| = \langle 1;0 \rangle \), the results \( b \) and \( \neg b \) according to (3) lead to \( \|b\| = \langle 1;1 \rangle \). It is a complete contradiction. It will also appear in the terminal fact named hypothesis.

The verification algorithm may look like this [24]:
1. Declare that rules are strictly true (by memorizing, if necessary, the truth-values set by the experts).
2. Establish the values truth of starting premises into one of valid combination of the strictly true and strictly false: \( \langle 1;0 \rangle \) or \( \langle 0;1 \rangle \).
3. Establish the truth of intermediate and terminal facts in \( \langle 0;0 \rangle \).
4. Perform a direct inference and obtain the truth of the conclusions (hypotheses).
5. Check the truth of hypotheses for admissibility.
6. If the truth of conclusion is inadmissible (for instance the truth vector is equal to \( \langle 0;0 \rangle \) or \( \langle 1;1 \rangle \)) then:
   6.1. Perform a back tracing of the inference to find out the source of the problem – the KB artifact.
   6.2. Eliminate the artifact (or add it to the appropriate list).
   6.3. Return to step 2
7. If not all possible input values of premises are checked then:
   7.1. Return to step 2.
8. Return the original truth values of the rules.
9. STOP!

There is a question about the procedure for setting the truth values of the start facts. In the limiting case, \( N \) start premises generate \( 2^N \) truth combinations. Even for a small KB containing a few tens or hundreds input facts, it leads to an explosive increase of the number of checks. To solve this problem, it is proposed to use the fact that each hypothesis is based on a relatively small number of start assumptions. Then for each hypothesis, you can select only the facts related to it. Since in real problems \( n \) is usually much smaller than \( N \), where \( n \) is the number of start facts of the hypothesis, we get a significant reduction in the computational load. The general scheme of the algorithm becomes the following:
1. Declare that rules are strictly true (by memorizing, if necessary, the truth values set by the experts).
2. Highlight a terminal fact (hypothesis).
3. Identify all the associated with hypothesis startup facts.
4. Establish the values of truth of starting facts are into one of valid combination of the strictly true and strictly false values: \( \langle 1;0 \rangle \) or \( \langle 0;1 \rangle \).
5. Establish the truth of intermediate and terminal facts in \( \langle 0;0 \rangle \).
6. Perform a direct inference and obtain the truth of the conclusion (hypothesis).
7. Check the truth of hypothesis for admissibility.
8. If the truth of conclusion is inadmissible (for instance the truth vector is equal to \( \langle 0;0 \rangle \) or \( \langle 1;1 \rangle \)) then:
   8.1. Perform a back tracing of the inference to find out the source of the problem – the KB artifact.
   8.2. Eliminate the artifact (or add it to the appropriate list).
   8.3. Return to step 4.
9. If not all possible input values of premises are checked, then return to step 4
10. If not all possible hypotheses are checked, then return to step 2.
11. Return the original truth values of the rules.
12. STOP!

4. Conclusion

In this paper, we consider the problem of checking the rule-based KB for internal anomalies in the form of contradictions and breaking of the inference by means of V\(^{TP}\)-logic. The advantage of the approach is the ability of such logics to remain operational in conditions of low veracity (uncertainty), falsity and inconsistency of assumptions. At the same time, the uncertainty or contradiction arising during the inference is projected onto a terminal fact (hypothesis), which can serve as a signal of trouble in the KB. This allows you to detect anomalies using the direct inference procedure for all valid combinations of the truth of input premises. To reduce the number of combinations, which may be exponentially large, it is proposed to consider sets of facts for each individual hypothesis, which significantly reduces the amount of calculations.

References

[1] Jiarratano J C and Riley G D 2004 Expert Systems: Principles and Programming, 4\(^{th}\) edn. (Course Technology)
[2] Gavrilova T A, Kudryavtsev D V and Muromtsev D I 2016 Knowledge engineering. Models and methods (SpB: Izdatelstvo «Lan’ ») (in Russian)
[3] Gegechkori E T and Chaadayev A V 2017 Modern problems of developing expert systems in the banking sector when reengineering business processes. International Journal of Applied and Fundamental Research 8(2) 195-197 (in Russian).
[4] Priyali Sharma. Expert system: meaning, classes and advantages. Available at: http://www.yourarticlerepository.com/management/mis-management/expert-system-meaning-classes-and-advantages-mis/70413/
[5] Davies T 1990 Knowledge Bases and Neural Network, in Synthesis in Artificial Intelligence in the Pacific Rim. Proceeding of the Pacific Rim International Conference on Artificial Intelligence (Nagoya) 717-722
[6] Golovko V A et al. 2018 Integration of artificial neural network and knowledge bases, Ontology of designing 8(3) 366-386 DOI: 10.18287/2223-9537-2018-8-3-366-386 (in Russian)
[7] Nguyen D Q et al. A Convolutional Neural Network-based Model for Knowledge Base Completion and its Application to Search Personalization. Available at: https://pdfs.semanticscholar.org/5b7b/be73153f383fa96de1218b6602d2ef3dc564.pdf
[8] Socher R, et al. Reasoning With Neural Tensor Networks for Knowledge Base Completion.
Available at: https://nlp.stanford.edu/pubs/SocherChenManningNg_NIPS2013.pdf

[9] Nguyen T A at al. 1987 Knowledge Base Verification, AI Magazine 8(2) 69-75

[10] Meseguer P and Preece A 1995 Verification and validation of knowledge-based systems with formal specifications, The Knowledge Engineering Review 10(4) 331-343 DOI: 10.1017/S0269888900007542

[11] Vermesan A A and Coenen F (eds.) 1999 Validation and Verification of Knowledge Based System (NY: Springer Science + Business Media) DOI: 10/1007/978-1-4757-6916-6

[12] Bindilatti A and da Silva A 2010 Verification and Validation of Knowledge Bases Using Test Cases Generated by Restriction Rules, International Journal of Artificial Intelligence and Expert Systems (IJAE) 3(4) 117-125

[13] Racunas S A, Shah N H and Fedoroff N V 2006 A case study in pathway knowledgebase verification BMC Bioinformatics 7 DOI: 10.1186/1471-2105-7-196

[14] Smirnov V V 2006 Methods and tools for knowledge bases verifying in integrated expert systems Thesis of the candidate of technical sciences (Moscow) (in Russian)

[15] Rybina G V and Smirnov V V 2005 Features of static anomalies detecting in knowledge bases containing certain types of NON-factors 3rd Int. Sci. and Pract. Seminar Integrated Models and Soft Computing in AI (Moscow: Fizmatlit) 358-363 (in Russian)

[16] Dolinina O N 2010 Classification of errors in knowledge bases of expert systems Vestnik of Saratov State Technical University 2 125-130 (in Russian)

[17] Pospelova L Ya and Chukanova O V Poisk protivorechiy v produktsionnykh bazakh znaniy (Search for contradictions in rule-based knowledge bases). Available at: http://www.vmest.ru/nuda/poisk-protivorechii-v-produktsionnykh-bazakh-znaniy/main.html. (in Russian).

[18] Proskuryakov D P 2014 Search for contradictions using a rule management strategy based on the domain ontology Proc. Baikal all-Russian conference "Information and mathematical technologies in science and management", Part III. (Irkutsk: ESI SB RAS) 166–170 (in Russian)

[19] Proskuryakov D P 2015 Conflict resolution managing in rule-based expert systems Vestnik of Irkutsk State Technical University 8 47-51 (in Russian)

[20] Arshinskiy L V 1998 Methods of processing of nonstrict propositions (Irkutsk: East-Siberian Institute of MIA of Russia) 40 (in Russian)

[21] Arshinskii L V 2007 Substantial and Formal Deductions in Logics with Vector Semantics, Automation and Remote Control 68(1) 139-148

[22] Arshinskii L V et al. 2017 Modeling of Complex Subject Areas Based on the Logics Vector Semantics Far East Journal of Mathematical Sciences (FJMS) 101(4) 813-823

[23] Arshinskiy L V, Ermakov A A and Nitezhuk M S 2019 Logic with Vector Semantic as a Means of Knowledge Bases Verification Ontology of designing 9(4) 510-521 DOI: 10.18287/2223-9537-2019-9-4-510-521 (in Russian)