PAPER • OPEN ACCESS

Verification of knowledge models by means of logic with vector semantics

To cite this article: L V Arshinskiy et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 760 012006

View the article online for updates and enhancements.

You may also like

- Ir 5*d*-band derived superconductivity in Laire
- A Bhattacharyya, D T Adroja, P K Biswas et al.
- Influence of anomalous truth values on logical inference in V^{1F}-logics as a basis for verification of rule-based systems L V Arshinskiy, V L Arshinskiy, M P Dunaev et al.
- <u>Charge segregation model for</u> <u>superconducting correlations in cuprates</u> <u>above T</u> E V L de Mello and J E Sonier

The Electrochemical Society Advancing solid state & electrochemical science & technology



DISCOVER how sustainability intersects with electrochemistry & solid state science research



This content was downloaded from IP address 18.191.84.32 on 05/05/2024 at 14:42

Verification of knowledge models by means of logic with vector semantics

L V Arshinskiy¹, M S Nitezhuk¹ and M P Dunaev²

¹Irkutsk State Transport University, Irkutsk, Russia ²Irkutsk National Research Technical University, Irkutsk, Russia

Abstract. The paper highlights the method of verification and identification of contradictions in the production knowledge bases using one class of logic with vector semantics, namely V^{TF} -logic. The general verification algorithm is described. It is noted that the use of such logics, while maintaining the need to iterate over the input facts, offers a procedure of direct inference and analysis of the inconsistency only in the case of conclusions-hypotheses, characterized by the value of truth named "complete contradiction". Tracing back the appropriate chain will identify and resolve the cause of the conflict.

1. Introduction

A knowledge model is one of the modern approaches to the description of subject areas (SA). The difference between knowledge-based models and traditional mathematical models is that the former describe not the subject area itself, but the idea that a person (expert) has about it. That is, a double reflection takes place: the subject area is reflected in the inner world of the expert, and this reflection, in turn, is reflected with certain knowledge models. This makes it possible to use knowledge modeling to describe almost any subject areas, one way or another perceived by man. This significantly distinguishes such models from traditional mathematical models, which work only where mathematical modeling is simple enough to use.

The main consumers of knowledge models today are computer programs called expert systems (ES). Precisely universality and "omnivory" of the knowledge approach to modeling suggest that the software based on it is "seriously and for a long haul", at least until other approaches embrace "the conceivable world" effectively enough to work with appropriate models.

The beginning of knowledge modeling as a separate direction, perhaps, should be counted off from the late '50s, in particular, from the attempt of Newell, Shaw and Simon to create a "General Problem Solver" (GPS) [1]. A distinctive feature of GPS as a software product was its division into two architectural components, which in modern interpretation are known as the knowledge base (KB) and the inference machine. KB contained a model of the subject area, the inference machine used the KB knowledge to obtain information that was not provided by the input (but could be inferred from KB). To some extent, these studies gave rise to a new at that time class of software – expert systems (ES). In Russia (USSR at that time), similar research was carried out in the field of situational control [2].

Today there are several basic ways of representing knowledge – knowledge models. These are semantic networks, frames, models based on logical calculus, primarily propositional calculus and first order predicate calculus, and production model [3]. The rest come down to them one way or another.

The most common model is production due to its external simplicity, modularity, ease of development and support. It is also important that from the algorithmization point of view the system of products

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI. Published under licence by IOP Publishing Ltd 1

"If..., then..." has an expressive force equivalent to the Turing machine, that is, it has the capabilities of a universal algorithmic model [4].

One of the most important stages of KB creation is verification. This is understandable – development errors reduce the value of the KB as a SA model. The constructed model can be incomplete, contradictory, and unreliable in some cases. The conclusions obtained with its help may contain errors. The problems of development of knowledge models are known as "NON-factors" affecting KB quality [5, 6]. At the same time, many authors consider the possible contradictory of knowledge as the main problem, since the presence of contradictions in the KB reduces the credibility of it as a model of SA.

How to detect contradictions? What kinds of contradictions can occur in the SA knowledge model? Today, these questions are most actively investigated for a productive approach to modeling.

In [7] the following types of contradictions are considered:

1) the internal contradiction of the rules, when elements of production mutually exclude each other, for example $a \rightarrow \neg a$ (here " \rightarrow " and " \neg " – symbols of implication and negation);

2) the contradictions between the two rules, which in reasoning leads to mutually exclusive conclusions, like *a* and $\neg a$;

3) internal inconsistency of chains of reasoning, as a result of which the facts contradicting the known ones can be added to the system;

4) contradictory of chains of reasoning that leads to the fact that the same set of data generates mutually exclusive situations.

The situation is further complicated by the fact that the inconsistency may not appear immediately. It can be hidden in words-synonyms [8, 9], in the features of the subject area [10, 11], etc. [12-14].

In our opinion, majority of traditional knowledge processing methods based on classical logical calculus and truth represented by two-element set $\{T, F\}$ (or $\{1, 0\}$) are poorly adapted for verification of knowledge base contradictory. This makes development of verification methods based on other logical approaches actual.

2. Logics with vector semantics. V^{TF}-logics

In [15] one of the non-classical logical calculus based on the concept of the truth vector was used to work with unreliable and contradictory information. Generally, it refers to an ordered tuple $\langle a_1;...; a_n \rangle$ from the interval [0, 1], each component of which describes the truth of a proposition relative to a certain aspect. This class of logics has received the name of logics with vector semantics [15, 16].

A separate component of a comprehensive assessment of the truth of the object *x* can be viewed as an aspect. For example, the truth of the statement "car *x* comfortable" can be evaluated in terms of engine power, ergonomics, suspension, etc. This point of view brings the vector of truth closer to a fuzzy vector of belonging *x* to a certain set (in this example, a set of comfortable cars). At the same time, categories of truth such as *Truth* or *Falsehood* also may be aspects. In this case, for example, two-dimensional vector logic with aspects (True; False) characterizes the proposition from the point of view of how close it is to Truth and Falsehood separately. The importance of each aspect is determined by its own set of evidences: pros and cons. It is this view and this representation of truth is used in so-called V^{TF} -logics (vector logics with aspects of Truth and Falsehood), which are special cases of logics with vector semantics and a convenient formal apparatus for identifying contradictions in knowledge models. An example of a software system using this approach is presented in [17–19].

3. Detecting contradictions

Different authors may have different interpretations of the contradictory/noncontradictory of the knowledge system. For instance, in [19], a system is considered to be noncontradict, if only admissible states are derived from admissible states, and in order to check the KB for consistency, it is necessary to know all admissible SA states. In [8, 9] it is said that a contradiction arises when KB has semantically incompatible elements. For example, incompatible pairs of properties or concepts. And so on. Despite of the apparent diversity of views, we proceed from the fact that a contradiction is the existence of two mutually exclusive judgments a and $\neg a$ that appear in the system due to any reason.

Then a check for inconsistency is a test of whether a valid SA state, being entered into the system in the form of a corresponding statement, can produce a situation of the form $a \& \neg a$ (& is a conjunction symbol). In the light of this, the consistency check by classical means may look as follows:

1) Input of an assertion corresponding to one of the possible SA states (conjunction of facts) into the system.

2) Verification of the statement on admissibility (non-zero truth value).

3) Inference step.

4) Check the updated statement (taking into account the result of the conclusion) for admissibility.

5) If the updated statement is valid and is not final (the next output step is possible), then return to step 3), else

6) If the updated statement is not valid, then the conclusion of the contradiction of the KB and STOP! Further analysis of the contradiction source should be fulfilled either by an expert, or by appropriate automated procedures.

7) If the statement is a goal and other states of SA are possible, then return to step 1, else make the conclusion on the consistency of the KB and STOP.

The process is repeated cyclically until all valid input states are searched through.

It is taken into account that the rules of inference generate only valid (true) judgments from valid (true) judgments. Inadmissibility, in the absence of false productions, is associated with the appearance of the conjunction $a \& \neg a$ (in one or another form).

We discuss that here the V^{TF}-logic makes its adjustments. The most significant one is that to detect inconsistencies it is not necessary to look for pairs of species *a* and $\neg a$. The contradictory of judgment *a* will represent itself by the vector of truth $\langle 1; 1 \rangle$. To do this, the logical inference procedure should include a stage of combining evidence, characteristic of a plausible, or rather – attached inference [20]. If the inference steps have results:

$$a_1, a_1 \rightarrow b - b$$
: $||b|| = \langle b^+_1; b^-_1 \rangle;$
 $a_2, a_2 \rightarrow b - b$: $||b|| = \langle b^+_2; b^-_2 \rangle;$

and the resulting truth values of the conclusion *b* is $||b|| = \langle b^+_1; b^-_1 \rangle$ and $||b|| = \langle b^+_2; b^-_2 \rangle$, the evidence is combined according to the rule:

$$||b|| = \langle b^{+}_{1} \oplus b^{+}_{2}; b^{-}_{1} \oplus b^{-}_{2} \rangle.$$
(1)

Here \oplus – triangular co-norm [21] in infix records. It has the property $x \oplus 1 = 1$. Examples are the well-known expression:

 $x \oplus y = x + y - xy$ and $x \oplus y = max(x, y)$.

As a result, if $\langle b^+_1; b^-_1 \rangle = \langle 1; 0 \rangle$ (strict true) and $\langle b^+_2; b^-_2 \rangle = \langle 0; 1 \rangle$ (strict false), association rule (1) gives the truth $\langle b^+; b^- \rangle = \langle 1; 1 \rangle$ – a complete contradiction. Having appeared at any step, the contradiction extends to all subsequent consequents. This suggests the possibility of using this inference to automate conflict detection procedures. The algorithm looks the following way:

1) Set the truth value of the input facts in the form of a valid combination of strict truth $\langle 1; 0 \rangle$ and (if necessary) strict false $\langle 0; 1 \rangle$.

2) Perform a direct output under the given truth values.

3) Check the truth of terminal facts (hypotheses).

4) If contradictory conclusions are found, find out the cause of the contradiction by reverse tracing the output chain, eliminate the contradiction and return to step 1.

5) Otherwise, if not all possible truth values of the input facts are checked, return to step 1.

6) Otherwise, no contradictions were found. STOP!

Therefore, control of contradictions in a KB using V^{TF} -logic involves setting all valid combinations of truth vectors of input facts, identifying conflicting hypotheses, revealing and resolving contradiction (contradictions) if they are detected. And the task is amenable to automation.

4. Conclusion

Verification of knowledge models is primarily a search for internal contradictions of the knowledge system. To detect a contradiction is to obtain a conclusion of the form $a\&\neg a$ from the admissible set of facts on the model used. This requires a search for all possible facts and all possible conclusions

derived from them with constant observation of the facts at each step of the inference, which makes the calculation procedure heavier. The use of the V^{TF} -logics, while preserving the need for too much input facts, provides the procedure for direct display and analysis of the inconsistency only in the case of receiving the findings-hypotheses, characterized by the truth value $\langle 1; 1 \rangle$ (complete contradiction). A reverse tracing of the corresponding chain will identify and resolve the cause of the conflict. This procedure looks computationally simpler.

5. References

- [1] Newell A, Shaw J C and Simon H A 1959 *Report on a General Problem-Solving Program* (RAND Corporation) p 27
- [2] Pospelov D A 1986 Situational control: theory and practice (Moscow: Science) p 288
- [3] Giarratano J C, Riley G D 2004 *Expert Systems: Principle and Programming* (Course Technology) p 856
- [4] Post E L 1943 Formal Reductions of the General Combination Decision Problem American J. of Mathematics. 65 197–215
- [5] Narinyani A S 1982 Underdetermined models and operations with underdetermined values Preprint no 400 of Computing Center of the Siberian Branch of the Academy of Sciencesof the USSR
- [6] Narinyani A S 1986 Underdetermined in the systems of knowledge representation and processing *Bull. of the Academy of Sciences of the USSR. Engineer. Cybernetics* **5** 3–28
- [7] Pospelova L Y and Chukanova O V 2009 The search for contradictions in the production knowledge bases *Information and telecommunication systems* 5 (Moscow: National Research Nuclear University) 23–7
- [8] Prockuriakov D P 2014 Search for contradictions with the help of production management strategy based on the ontology of the subject area *Proceedings of the XIX Baikal All-Russian conference "Information and Mathematical Technologies in Sci. and Control"* part III (Irkutsk: ISEM SB RAS) p 166–70
- [9] Proskuriakov D P 2015 Conflict resolution management in rule-based expert systems *Proceedings of Irkutsk State Technical University* **8(103)** 47–51
- [10] Arshinsky V L and Proskuriakov D P 2016 Application of ontologies and case-based reasoning to processing context in event modeling in energy security research *Modern technologies*. *System analysis*. *Modeling* 4(52) 94–100
- [11] Dolinyna O N 2010 Classification of errors in the knowledge bases of expert systems Bull. of SSTU 4(50)2 125–30
- [12] Antoniou G, van Harmelen F, Plant R and Vanthinen J 1998 Verification and Validation of Knowledge-Based Systems AI Magazine 19(3) 123–26
- [13] Zhang D 1993 Knowledge Base Verification: Issues and Approaches AAAI Technical Report WS-93-05 pp 148–49
- [14] Sun Y, Wu T, Li X and Guizani M 2017 A Rule Verification System for Smart Buildings *IEEE Transactions on Emerging Topics in Computing* **5(3)** 367–79
- [15] Arshinskii L V 2007 Substantial and formal deductions in logics with vector semantics Automation and remote control **68(1)** 139–48
- [16] Arshinskiy L V 2007 Mathematical models for handling incomplete and inconsistent data on the basis of logics with vector semantics *Modern technologies*. *System analysis*. *Modeling* 4(16) p90–9
- [17] Arshinskiy L V, Arshinskiy V L, Khomenko A P, Kargapol'tcev S K, Korol'kov B P and Aslamova V S 2017 Logical-mathematical modeling of complex subject areas based on the logics vector semantics *Far East J. of Mathematical Sci. (FJMS)* **101(4)** pp 813–23
- [18] Arshinskiy L V, Arshinskiy V L, Dunaev M P, Dunaev A M and Aslamova V S 2017 Development of the expert system of electrical equipment setup based on logic with vector semantics Far East J. of Electronics and Communications (FJEC) 17(5) pp 1119–25

- [19] Arshinskiy L V 2012 Application of relational tables in the development of knowledge bases to illstructured subject areas with incomplete and contradictory information *Modern technologies*. *System analysis. Modeling* 1(33) 90–9
- [20] Tepandi J 1990 Comparison of Expert System Verification Criteria: Redundancy *Proc. ECAI 90 Conf. Stockholm* pp 49–62
- [21] Handbook on the Artificial Intelligence vol 2 Models and Methods 1990 ed. D A Pospelov. (Moscow: Radio and communication) p 304
- [22] Klement E P, Mesiar R and Pap E 2000 *Triangular norms. Trends in Logic* vol 8 (Springer, Dordrecht)