British Journal of Applied Science & Technology
14(6): 1-6, 2016, Article no.BJAST.22737
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN ***international***
www.sciencedomain.org

Production-Tabular Knowledge Base Correctness Checking Tools

Robert Karayev[1*], Natella Sadikhova[1] and Khatira Imamverdiyeva[1]

[1]Institute of Control Systems of the National Academy of Sciences of Azerbaijan, Az1141, 9, B. Vahabzade str., Baku, Azerbaijan.

Authors' contributions

This work was carried out in collaboration between all authors. Author RK designed the study, performed the statistical analysis, wrote the protocol, wrote the first draft of the manuscript and managed literature searches. Authors NS and KI managed the analyses of the study and literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/22737
Editor(s):
(1) Wei Wu, Applied Mathematics Department, Dalian University of Technology, China.
Reviewers:
(1) Ankur Singh Bist, Uttar Pradesh Technical University, India.
(2) Fatih Basciftci, Selcuk University, Turkey.
Complete Peer review History: http://sciencedomain.org/review-history/13568

Short Research Article

Received 23rd October 2015
Accepted 19th February 2016
Published 5th March 2016

ABSTRACT

Production-tabular knowledge bases are widely used in commercial expert systems. One of the main problems arising from the operation of such knowledge bases is their correctness. The reliability of the inference mechanism and the robustness of the expert system at "paradigm shift" depend largely on successful resolution of this problem. The paper gives a formal definition of "correctness" of extended entry production-tabular knowledge bases and proposes an algorithm to control their correctness. The obtained results create theoretical preconditions to ensure the reliability and robustness of the production-tabular technologies widely used in expert systems of diagnostics, monitoring, management, forecasting, decision-making.

Keywords: Production-tabular knowledge bases; correctness; correctness checking tools.

**Corresponding author: E-mail: karayevr@rambler.ru;*

1. INTRODUCTION

Production-tabular knowledge bases under investigation are a class of hybrid structures of knowledge representation, in which production systems are described in terms of extended entry decision tables [1-3]. The formalism of decision tables can significantly expand the application of popular expert systems based on production rules.

In particular, they can be used to check the correctness (completeness and consistency) of knowledge bases, which is a critical issue for these systems.

The paper considers production-tabular knowledge bases, which are a class of hybrid structures of knowledge representation, in which production systems are described in terms of extended entry decision tables [1-3].

Even the earliest production expert systems TEIRESJAS [4], EMYCIN [5], KAS [6], IKEE [7] featured knowledge base verification tools.

The problem of the correctness of production knowledge bases has been discussed in a number of theoretical works [8-13].

However, said tools of control are generally focused on too closed and static problem environment models, i.e. on the situations, in which inference mechanisms are determined at the stage of their formation and are not corrected during operation.

This paper discusses tools for testing the correctness both at the stage of formation of inference mechanisms and at their possible modification during operation.

Tools of this kind are relevant for expert systems operating in open and dynamic problem environments typical for the vast majority of real-life problems.

The paper attempts to solve this issue in the framework of the mathematical model of decision table, regarded as the isomorphism of the production-tabular knowledge base.

For the analysis of correctness, we use the modified technique developed in [14,15] for limited entry decision tables (tables with double-digit “Yes / No” terms).

2. BASIC CONCEPTS AND DEFINITIONS

Formally, the decision table is given [14] by the set $T = \left\langle C, D, \tilde{C}, \tilde{D} \right\rangle$, where

- $C = \{C_i\}, i=\overline{1,m},$ is the set of conditions or identifiers of conditions regarded as the coordinates of a set of data vectors representing elementary states of the problem environment;

- $D = \{D_r\}, \ r=\overline{1,k}$, is the set of solutions or identifiers of solutions considered as coordinates of any totality of solution vectors;

- $\tilde{C} = \left\| c_{ij} \right\|, \ i=\overline{1,m}, \ j=\overline{1,n}, \ \tilde{D}=\left\| d_{rj} \right\|, \ r=\overline{1,k}, \ j=\overline{1,n+1}$ are the matrices establishing the relationship between the data vectors (or states) and solutions.

The general structure of a decision table is shown in Table 1.

Table 1. The general structure of a decision table

Table name	**Rule**				
	1	**2**	**…**	***n***	
Condition 1	A	B	…	C	← Condition values
Condition 2	Y	Y	…	N	
…	…	…	…	…	
Condition *m*	Y	N	…	…	
Solution 1	X	X	…		← X for each solution
Solution 2		X	…		
...			…	X	
Solution *k*	X		…		

The pair $R_j = \left\langle \widetilde{C}_j, \widetilde{D}_j \right\rangle$, $j = \overline{1,n}$, where $\widetilde{C}_j, \widetilde{D}_j$ are the vector-columns of matrices $\widetilde{C}$ and $\widetilde{D}$ is called the solutions rule (rule *R*).

The pair $E = \left\langle * , \widetilde{D}_{n+1} \right\rangle$, where the symbol " * " means that the first element of the pair is undetermined, is called the "otherwise" rule (rule *E*).

Rule *E* is used for fixing the situations which are anomalous in terms of semantics of the problem environment, and entered into a decision table to rectify possible incompleteness of the knowledge base.

A set of states of the problem environment is the set consisting of the data vectors $s_q = (C_i^q)$, $i = \overline{1,m}$,

where $C_i^q \in \hat{C}_i$; $q = \left\{ 1, 2, ..., \prod_{i=1}^{m} \left| \hat{C}_i \right| \right\}$.

Matrices $\widetilde{C} = \left\| c_{ij} \right\|$, $i = \overline{1,m}$, $j = \overline{1,n}$, $\widetilde{D} = \left\| d_{rj} \right\|$, $r = \overline{1,k}$, $j = \overline{1,n+1}$,

where $c_{ij} \in \left\{ \lambda \cup \hat{c}_i \right\}$, $d_{rj} \in \left\{ 0, 1, ..., k \right\}$ establish the relationship between the data vectors (or states) and solutions.

The values of the matrix elements $\widetilde{C}$ and $\widetilde{D}$ have the following meaning:

$$c_{ij} = \begin{cases} c \in \hat{C}_i, \text{ if the condition } C_i \text{ for the rule } R_j \text{ is } C \\ \lambda, \quad \text{if the condition } C_i \text{ for the rule } R_j \text{ is} \\ \quad\quad \text{immaterial}; \end{cases}$$

$$d_{rj} = \begin{cases} d \in \left\{ 1, 2, ..., k \right\}, \text{ if the decision } D_r \text{ is safisfied for} \\ \quad\quad \text{the rule } R_j \text{ } (E \text{ rules, if } j = n+1) \\ \quad\quad \text{and has priority and the execution order } d; \\ 0, \quad \text{if the action } D_r \text{ is not performed} \\ \quad\quad \text{for rules } R_j \text{ } (\text{rule } E, \text{if } j = n+1) \end{cases}$$

Usually, the elements $d_{ri} = 0$ are assumed to be "default" and not recorded in the decision table, and instead of the elements $c_{ij} = \lambda$, the symbol "–" is put.

Definition 1. A decision table that does not contain rules *E* is called *complete* relative to *S*, if $(\forall s_q \exists R_j)(s_q \to R_j)$.

Otherwise, a decision table is called *incomplete* relative to *S*.

Definition 2. A decision table is called *consistent* relative to *S*, if

$$(\exists s_q, R_j, R_p)\left[(s_q \to R_j) \,\&\, (s_q \to R_p) \Rightarrow (\widetilde{D}_j = \widetilde{D}_p) \right].$$

Accordingly, a decision table is called contradictory relative to *S*, if

$$(\exists s_q, R_j, R_p)\left[(s_q \to R_j) \,\&\, (s_q \to R_p) \,\&\, (\widetilde{D}_j \neq \widetilde{D}_p) \right].$$

In this case, we say that the data vector causes inconsistency of decision tables for rules R_j and R_p.

Definition 3. A decision table is called *correct* concerning *S*, if it is complete and consistent relative to *S*. Otherwise, a decision table is called *incorrect* relative to *S*.

The correctness of a decision table relative to *S* is also called semantic correctness or correctness relative to a given problem interpretation.

Definition 4. By a set of syntactically possible (assuming independence conditions C_i) situations *N,* we understand a set consisting of the data vectors

$$s_q = (C_i^q),\ i = \overline{1,m},\ q = 1, 2, ..., \prod_{i=1}^{m} \left| \hat{C}_i \right|.$$

The correctness of a decision table relative to the set *N* is the syntactic correctness or correctness relative to any problem interpretation.

Before proceeding to describe the correctness checking algorithms, we will make some remarks.

Remark 1. Since *S* is determined by the specifics of a current problem and is usually given implicitly (through a system of constraints), let us take the set *N* as *S* for universality.

Accordingly, will check the correctness of decision tables relative to the set *N*.

Remark 2. In the event of inconsistency or incompleteness of a decision table against *N*, we assume that there is a processor (e.g., the compiler of decision tables), capable of establishing the correctness or incorrectness of decision tables relative to *S* based on the output of the algorithm.

Thus, the issue of semantic correctness in this case depends on the processor.

3. CONSISTENCY CHECK

Let $R^k \subseteq R$. Vectors of conditions S^k of rules R^k form the matrix $\widetilde{C}^k = \|c_{ij}\|$, $i=\overline{1,m}$, $j \in J_k$, where J_k is a set of indexes of the rules included in R^k.

Definition 5. Vectors of conditions S^k will be called *equivalent* ("~"), if in each row of the matrix $\widetilde{C}^k$, all the essential elements $(c_{ij} \neq \lambda)$ are equal to each other or all of the elements except one are not essential $(c_{ij} = \lambda)$.

Accordingly, the combination of equivalent vectors S^k will be called an equivalent combination and labeled $\hat{K}$.

Lemma. To make a decision table consistent relative to *S*, it is necessary and sufficient that the relationship $(\forall jp)\left|(S_j \sim S_p)\&(\widetilde{D}_j = \widetilde{D}_p)\right|$, $j \neq p$. hold.

Accordingly, the necessary and sufficient condition of inconsistency of a decision table relative to *S* for the rules R_j and R_p is the relationship $(S_j \sim S_p)\&(\widetilde{D}_j \neq \widetilde{D}_p)$, $j \neq p$.

The scheme for the proof of the lemma is borrowed from [15].

Corollary 1. A decision table is consistent relative to *S* for rules R_j and R_p if at $\widetilde{D}_j = \widetilde{D}_p$, the essential elements $(c_{ij}, c_{ip} \neq \lambda)$ in at least one of the rows are not equal to each other $c_{ij} \neq c_{ip}$, $j \neq p$.

Corollary 2. The totality of data vectors that cause inconsistency of the decision table relative to *S* for rules R_j and R_p is determined by the pair $\langle S_j \sim S_p, D_i \neq D_p \rangle$; for each pair, the number of vectors causing inconsistency is $\prod_{i \in I_\beta} |\hat{c}_{ij}|$, where I_β is the row indexing set, in which both elements are not essential $(c_{ij}, c_{ip} = \lambda)$.

Consistency check algorithm is determined.

4. COMPLETENESS CHECK

According to Definition 1 (with the substitution of *N* for S), a decision table is complete relative to *N*, if $N \subseteq S^1$. Strict inclusion means that there are some non-empty intersections of elements from S_i^1. A decision table is then called *redundant* relative to *N*.

Check for completeness of a decision table (containing no rule *E*) will be carried out by comparing the number of solution rules presented in the table with *H* number of syntactically possible solution rules.

Proposition 1. $H = \prod_{i=1}^{m} |\hat{C}_i|$, where $\hat{C}_i$ is the set of values of conditions C_i.

Proposition 2. $G = \sum_{z=2}^{u} (-1)^z B(z), u \leq n$, where *B(z)* is the number of data vectors contained in various intersections of the elements from S^1 to *Z*, or the number of data vectors satisfying simultaneously *Z* solution rules.

Corollary 3. A decision table is complete relative to *N*, if *F* – *G* = *H*; incomplete relative to *N*, if *F* – *G* < *H*; redundant relative to *N*, if *F- G >H*.

Let us now give a method for calculating *B(z)* from the matrix $\widetilde{C}^k$. Using the principle of mathematical induction and the lemma, we can prove the following theorem.

Theorem. For there to be such a data vector s_q that ensures $(\forall i)(s_q \to R_{ji}), R_{ji} \in R, i \in \{1, 2, ..., z\}, z \leq n$, it is necessary and sufficient to satisfy the relationship $(S_1^k \sim ... \sim S_j^k \sim ... \sim S_z^k)$, where $K_z = \{K\}$ is the set of combinations from *n* to *t* vectors $|K_z| = \binom{z}{n}, z \leq n$, S_j^k is j-th vector-column of *k*-th combination of the vectors $S_j, k \in K_z$.

Completeness check algorithm is determined.

5. CONCLUSION

The isomorphism between decision tables and production structures gives grounds for regarding the proposed scheme of correctness control as basic for production-tabular systems in general, both for limited-entry and extended-entry systems.

It should also be noted that the scheme can be used both at the stage of development of production-tabular systems, and their possible modifications during operation. This is important in open and dynamic problem environment characterized by high requirements to reliability and promptness of decisions.

The proposed verification algorithm was used in the “System of reactive diagnostics of Ethernet LAN” [16], “System of on-line diagnostics of power plants” [17], and in the “System of predicting the preservation of sinus rhythm after the elimination of a cardiac fibrillation” [18].

ACKNOWLEDGEMENTS

The authors express gratitude to all anonymous reviewers for their helpful comments, which contributed to a broader presentation of the history of the issue and possibilities of application of the research results.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Russell S, Norvig P. Artificial intelligence: A modern approach. N.J.: Prentice Hall; 1997.
2. Hamby E. Programming decision tables. N.Y.: J. Wiley & Sons Inc; 1976.
3. The current status of expert system development and related technologies, IEEE Expert. 4; 1995.
4. Davis R. Interactive transfer of expertise. Artificial Intelligence. 1979;12(2). [TEIRESIAS].
5. [EMYCIN] Van Melle W, Shortliffe EA, Buchanan BG. EMYCIN. A knowledge engineer’s tool for constructing rule-based expert systems. In Buchanan BG, Shortliffe EH. Rule-Based Systems. Addison-Wesley, USA; 1984.
6. [KAS] Reboh R. Knowledge-Engineering techniques & tools in the prospector environment SRI Technical note 243. Stanford Research Ins; 1981.
7. [IKEE] Ludvig A, Mellis W, Tomas L. IKEE – An integrated knowledge engineering environment for rule-based development. Proc. of the 7th Int. Workshop on Expert Systems and Their Applications, France; 1987.
8. Puuronen S. Towards help for consistency checking during the knowledge programming of a rule base. Rept. of the 10th Semin., Tampere; 1987;Pt. 2.
9. Macs R, Van Dijk JE. On the role of ambiguity and incompleteness in the design of decision tables and rule-based systems. Compot. J. 1988;31(6).
10. Cragunt BJ. A decision-table-based processor for checking completeness and consistency in rule-based expert systems. Int. J. Man-Machine Studies. 1987;26(5).
11. Jan Vanthienen. Modeling decision table structures. Business Rules Journal. 2010;11(12).
12. Mitsel AA, et al. Diagnostics of models of knowledge bases on the basis of production systems. Bulletin of the Tomsk State University. 2012;32(5). (In Russian).
13. Turban E, Sharda R, Delen D. Decision support and business intelligence systems. (9th ed.). N.J: Prentice International Hall, USA; 2011.
14. Pollack S. Decision tables: Theory and practice. N.Y.: J. Willey & Sons Inc; 1971.
15. King P. Ambiguity in limited entry decision tables. Comm. ACM. 1986;2(10):21-29.
16. Mohamed Gamal IO. Reference maps for the reactive diagnosis of LAN Ethernet. Intern. Proc. of the Conf. on Telecommunications Technology. Minsk: Belarus State University; 2012. (In Russian).
17. El-Sayed AF. The application of diagnosis knowledge-based system to power plant operations. Proc. of the Intern. Power Systems Conf. Karlsruhe, Germany; 2011.

18. Boytcov S. New technologies in diagnosis and prevention of cardiovascular risk. Proc. of the Conf. on New Opportunities in the Diagnosis, Treatment, and Reduction in Mortality Due to Cardiovascular Diseases. Moscow; 2010. (In Russian).

© 2016 Karayev et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/13568