Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

UDC 004.832
DOI: https://doi.org/10.30837/ITSSI.2020.13.077

S. SHAPOVALOVA

GENERATION OF TEST BASES OF RULES FOR THE ANALYSIS OF PRODUCTIVITY OF LOGICAL INFERENCE ENGINE

The **subject** of research in the article are test tasks to determine the performance of logical inference engines based on rules. The **purpose** of the work is to create a method of forming a database of rules and a set of data for analyzing the performance of logical inference mechanisms according to the given characteristics of rule activation and the complexity of finding a way to the target conclusion. The article solves the following **tasks**: determining the requirements for the knowledge base to be formed; creation of a knowledge base model; creating a way to form rules; identifying ways to increase the number of rules; providing testing of logical inference mechanisms for the proposed test problem. The following **methods** are used: methods of comparison with the sample, graph theory, logical programming. The following **results** were obtained: the method provides opportunities: creation of conditions of rules that complicate the data flow network of the Rete-algorithm as much as possible; formation of test bases of rules for derivation both on logic of the first order, and on offer logic; simply increase the number of knowledge base rules while maintaining the output logic. The formation of the knowledge base is based on a graph that represents the meta-rules of mixing paints to obtain a new colour. The vertices of the graph are colour classes. Each metarule is either an edge leading to the OR vertex or a set of edges in the case of the AND vertex. Each meta-rule specifies a scheme for creating several rules, because its structural components are classes of paints. The given structure of the graph significantly complicates the logical inference, because to prove the truth of the conclusion on AND-vertices it is necessary to have the conclusions obtained in the previous steps of different search directions. Examples of rule formation are given. Target vertices are defined, which determine the simplest and most complex cases of logical inference. **Conclusions**: it was proposed a semantic model of the knowledge base in the form of AND/OR-graph, which allows you to test the effectiveness of the implementation of conflict resolution strategies, as well as heuristic algorithms; a method of creating tests for inference mechanisms, which allows you to generate a database of rules and a set of data of certain sizes, as well as to model the complexity of finding the target output and activating the rules. Ways to increase the number of rules of the knowledge base to complicate the problem of logical inference have been presented; formulation of tests to determine the performance of logic output mechanisms for the proposed test problem has been done.

Keywords: inference engine; performance analysis (benchmarking); expert systems (rule-based system); Manners; Waltz; Palette.

Introduction

Modern software needs an intellectual component, in particular, drawing conclusions based on pre-accumulated information and operational data. In practice, such a problem often has to be solved on the basis of the results of image analysis from cameras by neural networks, for example, in on-board software systems, in monitoring and diagnostic systems. Another application of rule-based systems is the analysis of large data sets, such as Web data.

The implementation of logical inference is carried out on the basis of software tools, which contains the mechanism of inference and resource representation of rules. If obtaining logical conclusions is one of the many tasks of the software system, its solution should not delay the execution of other tasks. On the other hand, in stand-alone expert systems with very large rule bases, speed is also an important factor and inference slows sharply with increasing task size.

Therefore, the development of new mechanisms of logical inference and improvement of existing ones, primarily aimed at optimizing the inference by speed. This criterion is also important when choosing a mechanism for inference to implement the current task. Thus, the development of tools for objective analysis of the performance of inference mechanisms is relevant and has practical significance.

The state of the issue

In [1], a framework for comparative analysis of inference engine based on rules (benchmarking) is proposed.

A world-renowned provider of high-performance rule-based systems, Production systems technologies currently provides the results of the effectiveness of the proposed software tools according to the Manners and Waltz tests [2]. These tests (benchmarks) are one of the historically first tasks of building rules for analyzing the performance of IE (inference engine) mechanisms. It is believed that Manners has proven itself to test the same IE on different platforms and processors [3]. This benchmark was first proposed in [4].

In [5], for comparative analysis of the efficiency of pattern matching algorithms TREAT and LEAPS, problems were identified that became classical IE tests. In addition to the mentioned Manners and Waltz, there are: WaltzDB, ARP, Weaver.

The article [6] proposes a methodology for computational experiments to compare the efficiency of expert system shells on the example of CLIPS, VP-Expert and Ibis. As a test task, simple knowledge bases were generated according to the instructions, which had no semantics and, according to the authors, could not be reference tests. In particular, the samples were submitted in terms of a given length.

In [7] the specified methodology and metrics of the analysis of productivity of means of development of expert systems are resulted.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative Technologies and Scientific Solutions for Industries. 2020. No. 3 (13)

Specific approaches to test problems are presented in [8, 9]. In [8] it is noted that the classical tests are focused on determining the speed of the output mechanisms by the logic of the first order, and proposed tests by the proposed logic: Chess and Christmas Tree.

In [9], optimized comparison algorithms are presented. Proof of their effectiveness was carried out directly on the graphs, using Petri net benchmark and The Object-to-Relational schema mapping benchmark. In the latter case, the mapping algorithms were tested on relational database schemas derived from UML classes.

Another way to determine the performance of IE is to create rules for processing existing large data sets. This is especially true of Semantic Web data. The project [10] created a resource OpenRuleBench, which presents a set of various tests to analyze the performance of systems. Test results for systems based on five different rule representation / processing technologies are presented. According to the technologies, the following software development tools are tested: 1) deductive databases: DLV, IRIS, Ontobroker; 2) Prolog-based systems: XSB, Yap; 3) production rule systems: Drools, Jess; 4) systems with triple engines: Jena, SwiftOWLIM, BigOWLIM; 5) general knowledge bases: CYC.

In [11], the Berlin SPARQL Benchmark (BSBM) test performed a comparative analysis of rule-based inference engines: Euler YAP Engine (EYE), Jena Inference Engine, BaseVISor. From the point of view of realization of considerations, the first of them (rule reasoner) is based on Prolog YAP engine, last two – on Rete. The BSBM test simulates an e-commerce usage scenario based on product data, suppliers, consumers, and product reviews. BSBM was chosen by the authors because it is closer to real enterprise scenarios than several independent datasets offered in OpenRuleBench.

Works [12-14] are also devoted to the problems of presentation and processing of Semantic Web knowledge. In these works, cases of logical inference based on rules that may be impossible (defeasible rules), in particular, in the presence of conflicting information. Such cases are presented in formal form and in the rules of logical languages used by software tools for processing information of Internet resources and ontologies (Semantic Web, Ontology Based Data). In [12], a method of generating knowledge bases for testing defeasible reasoning tools was proposed. The following tools were tested: ASPIC +, DEFT, DeLP, Flora-2, SPINdle. For computational experiments, logic derivation schemes in the form of chains, trees, directed acyclic graphs (chain, tree, directed acyclic graph) are proposed. The rules were represented by abstract classes of existential rules.

In these models of test problems, the search for the target conclusion is carried out either on graphs of state space (State Space Graphs), or on a fairly simple in structure schemes. The complexity increases due to the increase in the number of parameters of the problem, i.e. the number of rules that define the edges of the search graph. To bring the test tasks closer to the real ones, in particular, it is necessary that the current conclusion is made on the basis of the conclusions obtained in the previous steps of different areas of search. To fulfill this condition, the logical inference should be performed on a clearly defined AND/OR-graph representation of meta-rules.

The mechanisms of both directions of logical inference draw conclusions based on representations on the logic of the first order (with limitations characteristic of the current approach). This allows you to create a model of a test problem (benchmark problem), in which you can display the specificity of different cases of inference and generate both a database of rules and a set of data with the ability to determine the characteristics of both components. Such a test task can become universal for output mechanisms.

The aim of the work is to create a method of forming a database of rules and a set of data for analyzing the performance of inference mechanisms according to the given characteristics of rule activation and the complexity of finding a way to the target conclusion.

To achieve this aim, it is necessary to solve the following tasks:

1. Define the requirements for the knowledge base to be formed.
2. Create a knowledge base model.
3. Develop a way to form rules.
4. Identify ways to increase the number of rules.
5. To provide statements of tests of mechanisms of logical inference on the offered test problem.

Requirements for a knowledge-based system

For further definitions we use the representation of systems on production rules.

The rules for reasoning are contained in the Knowledge Base. Logical expressions that represent the conditions of derivation in the rules correspond to the logic of the first order. Specified and proven facts are accumulated in a special structure – working memory WM (Working Memory). Each step of logical inference is carried out on the basis of one of the rules applied in the case of proving the truth of its conditions and leads to the determination of new facts or refutation of previously proven.

WM working memory contains a display of the current state of the solution of the output problem in the form of a set of true facts F_i (Fact):

$$WM = \{F_1, F_2, \dots, F_n\}. \quad (1)$$

Each rule – or product – contains two parts, which, due to their location in the record of the j-th rule R_j (Rule), are denoted by LHS_j (Left-Hand Side) and RHS_j (Right-Hand Side), respectively:

$$LHS_j -> RHS_j, \quad (2)$$

Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

where LHS_j is a conditional part of the *j*-th production rule, RHS_j is a part of the *j*-th production rule.

Part LHS_j is a logical expression from samples P_k (Pattern), the structure of which corresponds to the facts of WM. Unlike facts, sample arguments can have unrelated variables. The RHS_j part contains a list of actions to change the WM. Each action means adding a new *F+* fact or removing an unnecessary for further reasoning *F*. All WM facts at each output step are compared with the P_k samples of the conditional parts of the rules.

According to the production model, the truth of a logical expression is proved by comparing the samples it contains with the facts of WM. So, the presence of all relevant facts proves the truth of the conjunctival connection. If the antecedent LHS_j is true, then the consistent RHS_j can be executed. In this case, the *j*-th rule is activated. This is its inclusion in the conflicting Agenda set. At each step of the conclusion of a particular strategy for resolving the conflict with the Agenda, one rule is chosen for firing. It is his part of the action that leads to the current changes in WM.

Thus, logical derivation in rule-based systems is based on two concepts: pattern matching and conflict resolution.

The most resource-intensive stage of logical inference is the activation of rules, which is carried out on the basis of comparison with the sample. Rule-based software development software tools mostly use the Rete algorithm for mapping. Therefore, when creating a model for forming a test knowledge base, the task was, on the one hand, to present the conditions in such a way as to use the conceptual advantages of the Rete algorithm to reduce comparison time, and, on the other hand, to complicate the data flow network.

According to the results of comparative analysis of Rete and TREAT comparison algorithms [15], the characteristics of KB rules are determined, which significantly affect the efficiency of logical inference. Based on them, the following requirements are set for the knowledge base to be formed according to the model:

1) the presence of temporary redundancy;

2) the presence of structural similarity;

3) the presence of the same variables in several samples of the conditional part of the *LHS* of all rules;

4) the presence of variables of the same name in the samples of the conditional part of *LHS* and the facts of *RHS* of all rules;

5) the number of conditional *LHS* elements of each rule is not less than 6;

6) the number of negative conditional elements of the *LHS* of each rule is not less than half of the total.

These requirements apply to KB rules. In addition, to track the logic of the output requires a semantic representation of KB, which provides:

7) the possibility of obtaining the same conclusion by different chains of reasoning;

8) modeling the process of logical inference based on conflict resolution strategies based on the principle of novelty.

The model of the knowledge base and the way of generating a set of data are abstract in the sense that they represent the concept of creating a test problem. Direct recordings of the representation of these components should be implemented in formats defined by inference mechanisms, the performance of which is analyzed.

Knowledge base model

The semantics of the Palette test knowledge base are based on the rules of mixing paints to obtain a new colour. This provides clarity for both KB views and tracing of the reasoning process.

Since the proposed model is intended for the formation of KB records, their representation has a single formalization. The representation uses the common notation of mathematics and logic of predicates, as well as the symbols <> to distinguish variable parts of names. Object names are used in Prolog syntax records, in particular, variable names start with a capital letter and atoms start with a lowercase letter.

We introduce the concept of colour classes. A colour class is a general designation of several colours that can be applied / obtained in the current logical output step. Original colours are represented by the classes *Primary*, *Secondary* and *Neutral*, which are defined as sets of such colours, respectively:

$$Primary = \{yellow, red, blue\}, \quad (3)$$

$$Secondary = \{orange, violet, green\}, \quad (4)$$

$$Neutral = \{white, black\}. \quad (5)$$

Classes *gray*, *brown* contain one corresponding colour. All other classes define colours after toning and have a name that is the first part of the name of the corresponding colour.

It is the colour classes that are the units of representation of meta-rules, for example:

$$PrimaryColour1\text{^}PrimeryColour2\text{->}$$
$$SecondaryColour, \quad (6)$$

where *PrimaryColourj* is a paint from the set (3), *SecondaryColour* is a paint from the set (4).

Fig. 1 shows the semantic model of KB in the form of AND/OR-graph.

The vertices of the graph are colour classes. Each meta rule is either an edge leading to the OR vertex or a set of edges in the case of the AND vertex. Fig. 1 indicates this by the corresponding marks. Each meta rule specifies a schema for creating multiple KB rules, because its structural components are paint classes. For example, according to meta-rule (6), 3 rules are created in the knowledge base, which determine the possibility of obtaining each colour from the set (4).

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative Technologies and Scientific Solutions for Industries. 2020. No. 3 (13)

In the table 1 in accordance with the symbols of fig. 1 are given the rules that determine the first step towards the goal.

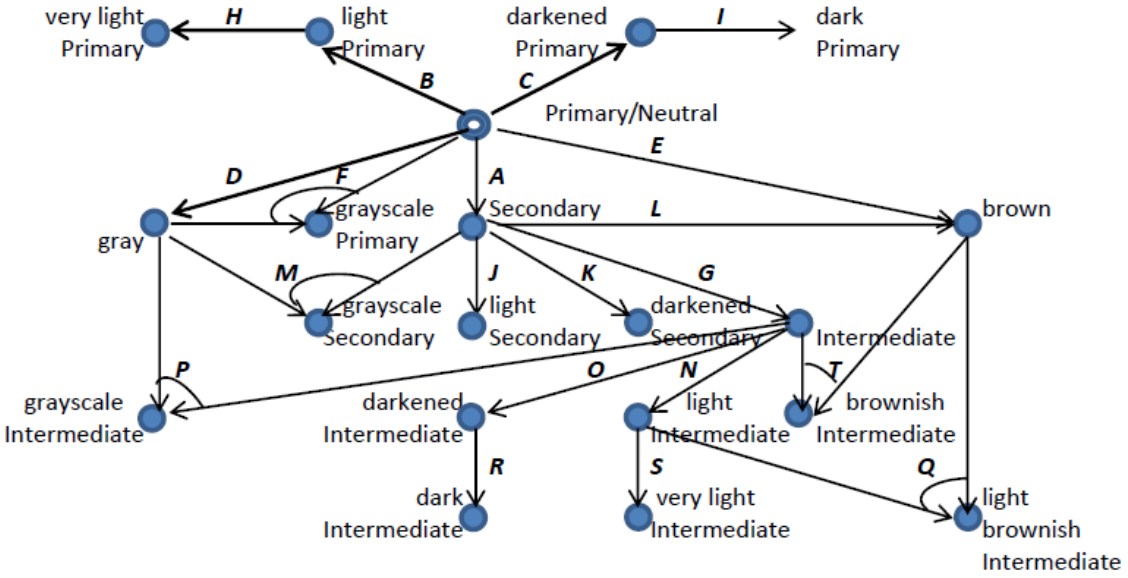

Fig. 1. AND/OR- graph representation of the Palette knowledge base

Table 1. *Defining the rules of the first step to the goal*

MetaRule		Rule	
Sign	**Scheme**	**#**	**Scheme**
A	*PrimaryColour1 ^ PrimeryColour2 -> SecondaryColour*	*1*	*red^yellow -> orange*
		2	*red^blue -> violet*
		3	*yellow^blue -> green*
B	*PrimaryColour ^ white -> lightColour*	*4*	*red^white -> light_red*
		5	*yellow^white -> light_yellow*
		6	*blue^white -> light_blue*
C	*PrimaryColour ^ black -> darkenedColour*	*7*	*red^black -> darkened_red*
		8	*yellow^black -> darkened_yellow*
		9	*blue^black -> darkened_blue*
D	*white ^ black -> gray*	*10*	*white ^black -> gray*
E	*PrimaryColour1 ^ PrimaryColour2 ^ PrimaryColour3 -> brown*	*11*	*red^yellow^blue -> brown*

Obtaining the colour *brown* can be achieved in two ways: by meta-rule *E* or by meta-rule *L* (fig. 1). Due to this, the requirement to obtain the same conclusion by different chains of reasoning is fulfilled. Meta rule *L* defines 3 rules for mixing colour from set (4) and colour from colour from set (4), which has not yet been used in the output process.

The colours of the *Intermediate* class on the palette are intermediate between the corresponding *PrimaryColour* and *SecondaryColour* and are created according to meta-rule *G* (table 2).

Table 2. *Defining the rules of the Intermediate class*

MetaRule		Rule	
Sign	**Scheme**	**#**	**Scheme**
G	*PrimaryColour ^ SecondaryColour -> IntermediateColour*	*12*	*red^violet -> red_violet*
		13	*red^orange -> red_orange*
		14	*orange^yellow -> orange_yellow*
		15	*green^yellow -> green_yellow*
		16	*blue^green -> blue_green*
		17	*blue^violet -> blue_violet*

Further rules are built in the same way. They have the idea of toning all the colours that can be obtained on the basis of initial data. Toning is either one of the specified neutral colours (5), or one of those obtained in the output process: *gray*, *brown*.

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13) ISSN 2524-2296 (online)

The name of the tinted colour *Toned* according to fig. 1 is an element of a subset of colour classes:

Toned $\in$ *{light, very_light, darkened, dark, grayscale, brownish}.* (7)

Toning is carried out according to the scheme:

Colour ^ *Tone* -> <*Toned*>_<*Colour*>, (8)

where *Colour* is any colour specified or obtained by mixing, <*Toned*> is the definition of the tinted colour from the set (7), <*Colour*> is the part of the name of the tinted colour that matches the name of the colour *Colour*.

Examples of determining the tinted colour (8) are presented in table 3.

Table 3. *Definition of tinted colour*

Available colour		Tinted colour	The resulting colour	
Marking	**Sample**		**Marking**	**Sample**
<*Colour*>	*red*	*white*	*light_<Colour>*	*light_red*
light_<Colour>	*light_red*	*white*	*very_light_<Colour>*	*very_light_red*
<*Colour*>	*red*	*black*	*darkened_<Colour>*	*darkened_red*
darkened_<Colour>	*darkened_red*	*black*	*darkened_<Colour>*	*dark_red*
<*Colour*>	*red*	*gray*	*grayscale_<Colour>*	*grayscale_red*
<*Colour*>	*red*	*brown*	*brownish_<Colour>*	*brownish_red*

The given structure of the graph significantly complicates the logical inference, because to prove the truth of the conclusion on AND-vertices it is necessary to have the conclusions obtained in the previous steps of different search directions. This provides additional opportunities to test the implementation of conflict resolution strategies.

The proposed model provides the concept of generating rule bases. The rule entry must match the syntax and format of the inference mechanism.

The way to present the rules

The knowledge base model reflects the sequence of application of rules in the process of logical inference. Each rule must be presented according to (2).

To represent knowledge usually use special declarative languages, which in many cases are adapted to a certain formalization of logical conditions. Let's define rules by the clause form. We will assume that the j-th phrase (clause) of such language represents a rule, the conditional part of which is a conjunctive connection of samples:

$$LHS_j = P_1 \wedge P_2 \wedge \ldots \wedge P_n, \quad (9)$$

where LHS_j is a conditional part of the j -th production rule, P_k is a k-th sample.

Let's determine the format of the samples. You can use two approaches:

1) create different formats for each level of the graph/branch of reasoning KB,

2) create samples of the same format, where the search depth is displayed in the name of the samples according to a given algorithm.

The following is the method of the latter approach. The minimum number of samples required to meet the above requirements 1-5 of the KB rules is presented.

The single LHS format for all rules contains 3 parts:

1) direct representation of colours that determine the conditions of transitions;

2) abstract conditions of "load", in which variables are introduced, most of which must be included in several samples;

3) "load" conditions associated with colours that contain variables are the same as in abstract "load" conditions.

Colour is represented by facts:

colour(Colour), (10)

where *colour* – the name of the predicate that identifies the pattern, *Colour* is the name of the colour.

To complicate the comparison, the common for all rules of one level AND/OR-graph "load" conditions are implemented, which contain positive and negative atomic formulas and an additional condition that limits the unrelated variable. For the first level, the "load" condition is such a logical connection:

f1(1,X) ^ *f2(X,Y,2)* ^ *Y>0* ^ ⌐ *f3(Y,3,X),* (11)

where *f1, f2, f3* are names of samples, *X, Y* – unrelated variables.

In the conditional part of the rules of each subsequent level in the functor is added one:

f<1>1(1,X)^f<1>2(X,Y,2)^Y>0 ^ ⌐ *(f<1>3(Y,3,X)),* (12)

where *f<1>1, f<1>2, f<1>3* are names of samples, *X, Y* – unrelated variables.

The *LHS* part of the rule contains only one connection (12). The link that corresponds to the colour obtained later than the other (that is, with more units in the sample names) is represented.

Additional "load" conditions associated with colours are represented by the following samples:

¬*f4_<Colour>(Z,z)*^¬*f5_<ColourName>(check,Z,Y,X),* (13)

where *X, Y, Z* - unrelated sample variables, *Colour* – name of colour, <*Colour*> – the part of the sample name that matches the colour name.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative Technologies and Scientific Solutions for Industries. 2020. No. 3 (13)

Thus, *LHS* is a conjunctive relationship of conditions and is determined by (10 – 13).

Running the rule (firing) means getting a new paint, i.e. adding to the WM 6 facts that correspond to the samples of the new colour obtained from the expressions (10 – 13). Part of the rule is a set of these facts:

$$RHS_j := \left\{ F_1^+, F_2^+, ..., F_6^+ \right\}, \quad (14)$$

where RHS_j – conditional part of the j -th production rule, F_1^+ – i- th fact that corresponds to one of the expressions (10 – 13).

In this method, the logical output is considered monotonous – the facts are only added to the working memory, deletion is not provided. Therefore, in the future the facts are recorded without a mark " + ".

Here are examples of typical rules of the knowledge base by formulas (2, 9-13) and:

1) meta-rule *A* (fig. 1):

$$colour(red) \wedge colour(blue) \wedge f1(1,X) \wedge f2(X,Y,2) \wedge Y>0 \wedge \neg f3(Y,3,X) \wedge \quad (15)$$
$$\wedge \neg f4\_red(Z,z) \wedge \neg f5\_red(check,Z,Y,X) \wedge \neg f4\_blue(Z,z) \wedge \neg f5\_blue(check,Z,Y,X)$$
$$\rightarrow$$
$$\{ colour(violet), f11(1,X), f12(X,Y,2), \neg f13(Y,3,X)), \neg f4\_violet(Z,z), \neg f5\_violet(check,Z,Y,X) \}$$

2) meta-rule *P*:

$$olour(red\_violet) \wedge colour(gray) \wedge \quad (16)$$
$$f111(1,X) \wedge f112(X,Y,2) \wedge Y>0 \wedge \neg(f113(Y,3,X)) \wedge$$
$$\wedge \neg f4\_red\_violet(Z,z) \wedge \neg f5\_red\_violet(check,Z,Y,X) \wedge \neg f4\_gray(Z,z) \wedge \neg f5\_gray(check,Z,Y,X)$$
$$\rightarrow$$
$$\{colour(grayscale\_red\_violet), f1111(1,X), f1112(X,Y,2), \neg f1113(Y,3,X), \neg f4\_grayscale\_red\_violet(Z,z),$$
$$\neg f5\_grayscale\_red\_violet(check,Z,Y,X)\}$$

All rules (except those created by meta-rule *E*) are a mixture of two paints, i.e. their *LHS* contains 9 samples (3 for each paint and 3 – "load"). For rule *E*, the *LHS* contains 12 samples.

The values of the set of variables {*ColourName, X, Y, Z*} are determined by the initial facts of WM, which correspond to the colour samples of classes (3, 5) (Primary/Neutral vertex).

Generating a data set of the required size is done by creating additional facts of working memory, which may or may not meet the samples of *LHS* rules.

Increasing the knowledge base

You can create 70 rules from the graph shown in fig. 1.

You can increase the knowledge base by adding new rules not applied in the presented AND/OR-graph. The deepest level on the way to the target conclusion is 4. Obviously, the graph can be completed at all previous levels. For example, add a meta rule of mixing *Primary* colours with brown. In addition, you can add new classes of shades, increasing the number of transitions to the target output.

Another approach to increasing the knowledge base with an unchanged AND/OR-graph structure is to define new *LHS* formats and generate rules based on them, which actually duplicate the existing rules with insignificant, in terms of records, differences. For example, you can leave the colour representation unchanged. It is enough to replace the symbol "f" (according to the formulas (11-12)) in the samples and facts of the rules with another symbol or their sequence and turn the previously generated rules into new ones. Thus, you can multiply the number of rules while maintaining the logic of inference by forming new parts of the knowledge base as a reflection of previously formed.

Benchmark Palette testing

To complicate the task of testing the performance of inference mechanisms based on Palette knowledge in the data set can be provided:

- the existence of several successful substitutions of the set of arguments of the conditional part of the rule {*X, Y, Z*} to create the appropriate number of instances (instantiation) of the same rule;
- increasing the number of variables *LHS* (when generating a data set for samples other than those proposed above);
- the presence of facts, the arguments of which do not agree with the arguments of the samples of the conditional part of any rule.

The given structure of the graph of the knowledge base (fig. 1) significantly complicates the logical inference. To prove the truth of the conclusion on AND-vertices, it is necessary to have the conclusions obtained in the previous steps of different search directions.

Thus, the complexity of modeling is determined by:

1) the depth of the target conclusion on the graph;
2) the presence and number of AND-vertices;
3) the presence of more than one path to the goal on the KB column;
4) the number and set of given facts that are not used on the way to the conclusion, but increase the activation

Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13) *ISSN 2522-9818 (print)* *ISSN 2524-2296 (online)*

time, search space and the corresponding structures of the Agenda and WM;

5) the presence of several solutions in the same way to the goal due to several successful substitutions of samples;

6) the presence of a set / sets of facts WM, which are successfully compared with the relevant samples, but their arguments of the same name do not agree with each other in the conditional part of the rules.

The first two criteria relate to the choice of the target conclusion. This provides additional opportunities to test the implementation of conflict resolution strategies.

All other criteria are taken into account when creating a data set.

The knowledge base generated on the basis of the proposed method provides an opportunity to test the mechanisms of logical inference:

1) directly when proving a given target conclusion.

In the simplest case, the target output is very *light_ <Primary>* or *dark_ <Primary>*, in the most *complex – light_brownish_ <Intermediate>* (fig. 1). Criteria for the complexity of the data set are taken into account when generating the initial state of working memory;

2) to determine the effectiveness at the stage of activation.

It is possible to determine the time of activation of rules separately for any stage of logical inference. In the most complex case, a mandatory part of the working memory is to present all the facts about the colours that can be obtained in the current output task, and all the facts that correspond to the levels of the graph. An additional part is generated to complicate the activation according to the above criteria 3-6;

3) to test the efficiency/correctness of heuristic algorithms.

The semantic model of the knowledge base allows you to create heuristics to determine the priority of Agenda rules in conflict resolution. Heuristics, the parameters of which are colours, will allow you to follow one branch of reasoning. Testing of heuristic algorithms can be performed as a search on the AND/OR-graph.

The proposed method of forming a knowledge base and data set for analyzing the performance of implementations of logical inference mechanisms has the following properties that distinguish it:

- the possibility of artificially creating the conditions of the rules that will complicate the data flow network (data flow network) Rete-algorithm;
- the ability to test algorithms for resolving conflicts, including heuristics;
- the ability to create a benchmark for output both by first-order logic and by propositional logic. (In the latter case, in the terms of the rules there are statements - colours (table 1-3));
- a simple way to increase the number of rules of the knowledge base while maintaining the logic of derivation by forming new parts of the knowledge base as reflections of previously formed.

Conclusions

1. The analysis of benchmarks for mechanisms of derivation of systems based on rules is carried out. It is determined that for an objective performance analysis, it is advisable to generate both a rule base and a data set with the ability to determine the characteristics of both components.
2. The analysis of researches on comparison of efficiency of mechanisms of logical inference and algorithms of comparison is carried out. The requirements for the representation of the conditions of logical inference in the rules are defined.
3. A semantic model of the knowledge base in the form of AND / OR-graph is proposed, which allows to test the effectiveness of the implementation of conflict resolution strategies, as well as heuristic algorithms.
4. A method for creating tests for inference mechanisms is proposed, which allows to generate a database of rules and a set of data of certain sizes, as well as to model the complexity of finding the target output and activating rules.
5. Ways to increase the number of rules of the knowledge base to complicate the problem of logical inference are presented.
6. Presentations of tests to determine the performance of logical inference mechanisms for the proposed test problem are presented.

References

1. Bobek, S., Misiak, P. (2017), "Framework for Benchmarking Rule-Based Inference Engines", *Artificial Intelligence and Soft Computing, ICAISC 2017, Lecture Notes in Computer Science*, Springer, Cham., Vol. 10246, P. 399–410. DOI: https://doi.org/10.1007/978-3-319-59060-8_36. SCOPUS.
2. "Clips R2 Benchmark" (2020), *Production systems technologies*, available at: http://www.grossweb.com/pst/clips-r2-benchmark/ (last accessed 25.09.20).
3. Riley, G. "Rearchitecting CLIPS" (2008), Business Rules Knowledge Base - October Rulefest 2008, available at: http://bizrules.info/conference/ORF2008DFW/GaryRiley_RearchitectingCLIPS_ORF2008.pdf (last accessed 25.09.20).
4. Kieman, G., de-Maindriville, C., Simon, E. (1990), "Making Deductive Database a Practical Technology: A Step Forward", *Institute National de Recherche en Informatique et en Automatique*, Report No. 1153, P. 237–245. DOI: https://doi.org/10.1145/93605.98733
5. Brant, D, Grose, T., Lofaso, B., Miranker, D. P. (1991), "Effects of Database Size on Rule System Performance: Five Case Studies", *Proceedings of the 17th International Conference on Very Large Data Bases*, P. 287–296.
6. Plant, R., Salinas, P. (1994), "Expert systems shell benchmarks: The missing comparison factor", *Information & Management*, Vol. 27, Issue 2, August 1994, P. 89–101. DOI: https://doi.org/10.1016/0378-7206(94)90009-4
7. Farinha, J. M. T. (2018), *Asset Maintenance Engineering Methodologies*, CRC Press, 322 p.
8. Hicks, R. C., Wright, K. (2009), "Performance Testing of Propositional Logic Inference Engines", *Journal of Computer Information Systems*, Published online, Vol. 49, P. 122–126.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative Technologies and Scientific Solutions for Industries. 2020. No. 3 (13)*

9. Bergmann, G., Horváth, Á., Ráth, I., Varró, D. (2008), "A Benchmark Evaluation of Incremental Pattern Matching in Graph Transformation", *Graph Transformations, ICGT 2008, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol 5214. P. 396–410. DOI: https://doi.org/10.1007/978-3-540-87405-8_27
10. Liang, S., Fodor, P., Wan, H., Kifer, M. (2009), "OpenRuleBench: An analysis of the performance of rule engines", *Proc. 18th International Conference on World Wide Web*, P. 601–610. DOI: https://doi.org/10.1145/1526709.1526790
11. Rattanasawad, T., Buranarach, M., Saikaew, K.R., Supnithi, T. (2018), "A Comparative Study of Rule-Based Inference Engines for the Semantic Web", *IEICE Transactions on Information and Systems*, January 2018, P. 82–89. DOI: https://doi.org/10.1587/transinf.2017SWP0004
12. Hecham, A., Croitoru M., Bisquert P. (2018), "A First Order Logic Benchmark for Defeasible Reasoning Tool Profiling", *RuleML+RR*, Sep 2018, Luxembourg, P. 81–97. DOI: https://doi.org/10.1007/978-3-319-99906-7_6
13. Hecham, A., Croitoru, M., Bisquert P. (2018), "Demonstrating a benchmark for defeasible reasoning", *Computational Models of Argument, ser. Frontiers in Artificial Intelligence and Applications*, Vol. 305, P. 461–462. DOI: https://doi.org/10.3233/978-1-61499-906-5-461
14. Hecham, A., Croitoru, M., Bisquert P. (2018), "On a flexible representation for defeasible reasoning variants", *AAMAS'18: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, P. 1123–1131.
15. Shapovalova, S., Mazhara, O. (2015), "Formalization of basic pattern matching algorithms in production systems", *Eastern European Journal of Enterprise Technologies*, No. 4/3 (76), P. 22–27. DOI: https://doi.org/10.15587/1729-4061.2015.46571.

Received 14.08.2020

Відомості про авторів / Сведения об авторах / About the Authors

Шаповалова Світлана Ігорівна – кандидат технічних наук, доцент, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", доцент кафедри автоматизації проектування енергетичних процесів і систем, Київ, Україна; email: lanashape@gmail.com; ORCID: http://orcid.org/0000-0002-3431-5639.

Шаповалова Светлана Игоревна – кандидат технических наук, доцент, Национальный технический университет Украины "Киевский политехнический институт имени Игоря Сикорского", доцент кафедры автоматизации проектирования энергетических процессов и систем, Киев, Украина.

Shapovalova Svitlana – PhD (Computer Sciences), Associate Professor, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Associate Professor of the Department of Automation of Designing of Energy Processes and Systems, Kyiv, Ukraine.

ГЕНЕРАЦІЯ ТЕСТОВИХ БАЗ ПРАВИЛ ДЛЯ АНАЛІЗУ ПРОДУКТИВНОСТІ МЕХАНІЗМІВ ЛОГІЧНОГО ВИВЕДЕННЯ

Предметом дослідження в статті є тестові задачі визначення продуктивності механізмів логічного виведення, що базуються на правилах. **Мета** роботи – створення методу формування бази правил та набору даних для аналізу продуктивності механізмів логічного виведення за заданими характеристиками активації правил та складністю пошуку шляху до цільового висновку. В статті вирішуються наступні **завдання**: визначення вимог до бази знань, що має формуватись; створення моделі бази знань; створення способу формування правил; визначення способів збільшення кількості правил; надання постановки випробовувань механізмів логічного виведення за запропонованою тестовою задачею. Використовуються такі **методи**: методи співставлення зі зразком, теорії графів, логічного програмування. Отримано наступні **результати**: метод надає можливості: створення умов правил, які максимально ускладнюють мережу потоку даних Rete-алгоритма; формування тестових баз правил для виведення як за логікою першого порядку, так і за пропозиційною логікою; простого збільшення кількості правил бази знань зі збереженням логіки виведення. Формування бази знань здійснюється на основі графу, який представляє метаправила змішування фарб для отримання нового кольору. Вершинами графу є класи кольорів. Кожне метаправило є або ребром, що веде до OR-вершини, або сукупністю ребер – у випадку AND-вершини. Кожне метаправило задає схему для створення декількох правил, оскільки його структурними компонентами є класи фарб. Надана структура графу значно ускладнює логічне виведення, оскільки для доведення істинності висновку на AND-вершинах необхідно мати висновки, отримані на попередніх кроках різних напрямів пошуку. Наведено приклади формування правил. Визначено цільові вершини, які визначають найпростіший та найскладніший випадки логічного виведення. **Висновки**: запропоновано: семантичну модель бази знань у вигляді AND/OR-graph, яка дозволяє випробовувати ефективність реалізацій стратегій розв'язання конфліктів, а також евристичних алгоритмів; метод створення тестів для механізмів логічного виведення, що дозволяє генерувати базу правил та набір даних визначених розмірів, а також моделювати складність пошуку цільового висновку та активації правил. Представлено: способи збільшення кількості правил бази знань для ускладнення задачі логічного виведення; постановки випробовувань з визначення продуктивності механізмів логічного виведення за запропонованою тестовою задачею.

Ключові слова: механізм логічного виведення (inference engine); аналіз продуктивності (benchmarking); експертні системи (rule-based system); Manners; Waltz; Palette.

ГЕНЕРАЦИЯ ТЕСТОВЫХ БАЗ ПРАВИЛ ДЛЯ АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ МЕХАНИЗМОВ ЛОГИЧЕСКОГО ВЫВОДА

Предметом исследования в статье являются тестовые задачи определения производительности механизмов логического вывода, основанного на правилах. **Цель** работы - создание метода формирования базы правил и набора данных для анализа производительности механизмов логического вывода по заданным характеристикам активации правил и сложности поиска пути к целевому заключению. В статье решаются следующие **задания**: определение требований к формируемой базе знаний;

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13) ISSN 2524-2296 (online)

создание модели базы знаний; создание способа формирования правил; определение способов увеличения количества правил; представление постановки испытаний механизмов логического вывода по предложенной тестовой задаче. Используются следующие **методы**: методы сопоставления с образцом, теории графов, логического программирования. Получены следующие **результаты**: метод предоставляет возможности: создания условий правил, которые максимально затрудняют сеть потока данных Rete-алгоритма; формирования тестовых баз правил вывода как как для логики первого порядка, так и пропозициональной логики; простого увеличения количества правил базы знаний с сохранением логики вывода. Формирование базы знаний осуществляется на основе графа, представляющего метаправила смешивания красок для получения нового цвета. Вершинами графа являются классы цветов. Каждое метаправило является или ребром, которое ведет к OR-вершине, или совокупностью ребер – в случае AND-вершины. Каждое метаправило задает схему для создания нескольких правил, поскольку его структурными компонентами являются классы красок. Представленная структура графа значительно усложняет логический вывод, поскольку для доказательства истинности вывода на AND-вершинах необходимо иметь выводы, полученные на предыдущих шагах различных направлений поиска. Приведены примеры формирования правил. Определены целевые вершины, которые определяют самый простой и самый сложный случаи логического вывода. **Выводы**: предложены: семантическая модель базы знаний в виде AND/OR-graph, которая позволяет испытывать эффективность реализаций стратегий разрешения конфликтов, а также эвристических алгоритмов; метод создания тестов для механизмов логического вывода, который позволяет генерировать базу правил и набор данных определенных размеров, а также моделировать сложность поиска целевого заключения и активации правил. Представлены: способ увеличения количества правил базы знаний для усложнения задачи логического вывода; постановка испытаний по определению производительности механизмов логического вывода по предложенной тестовой задачей.

Ключевые слова: механизм логического вывода; анализ производительности; экспертные системы; Manners; Waltz; Palette.

Бібліографічні описи / Bibliographic descriptions

Шаповалова С. І. Генерація тестових баз правил для аналізу продуктивності механізмів логічного виведення. *Сучасний стан наукових досліджень та технологій в промисловості*. 2020. № 3 (13). С. 77–85. DOI: https://doi.org/10.30837/ITSSI.2020.13.077.

Shapovalova, S. (2020), "Generation of test bases of rules for the analysis of productivity of logical inference engine", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (13), P. 77–85. DOI: https://doi.org/10.30837/ITSSI.2020.13.077.