An Algorithm to Compare Computer-security Knowledge from Different Sources

Gulnara Yakhyaeva and Olga Yasinkaya
Department of Information Technology, Novosibirsk State University, Novosibirsk, Russian Federation

Keywords: Information Security, Cyber Threats, Case-based Model, Fuzzy Model, Generalized Fuzzy Model, Generalized Case.

Abstract: In this paper we describe a mathematical apparatus and software implementation of a module of the RiskPanel system, aimed to compare computer-security knowledge learned from various online sources. To describe this process, we use model-theoretic formalism. The knowledge of a particular computer attack obtained from the same source is formalized as an underdetermined algebraic system, which we call a generalized case. The knowledge base is a set of generalized cases. To implement the knowledge comparison, we construct a generalized fuzzy model, the product of all algebraic systems stored in the database.
We consider an algorithm for computing consistent truth values and describe a software implementation of the developed methods. The developed algorithm has polynomial complexity.

1 INTRODUCTION

The main problem in designing intelligent systems is how to represent and process knowledge. Computer programs should have knowledge of a given subject domain presented in a formalism that is useful for the program. Knowledge representation consists mainly of identifying the most appropriate formalisms for representing knowledge and the most effective methods for manipulating this knowledge (Thayse, 1989).

This problem is particularly acute for knowledge of information security and cyber threats. In these subject domains, the value of information depends much more on its novelty than in most other scientific and technological domains. To effectively protect against computer threats, they must be identified as early as possible. Text in natural language on the Internet is one of the most relevant sources of such information. This gives rise to the need of representing security knowledge as ontologies. There are many application of knowledge based systems to computer security (for example (Ruhroth et al., 2014), (Gartner et al., 2014), (Burger et al., 2013)).

One method to process knowledge learned from text in natural language is model-theoretic knowledge representation, based on the model-theoretic approach developed to formalize domain ontologies (Palchunov, 2008) and Case-based reasoning methodology (Kolodner, 1992), (Assali et al., 2013). Under this approach, the knowledge learned from texts written in natural language is presented as algebraic systems (domain ***cases***) (Yakhyaeva and Yasinskaya, 2014). Using these systems, a ***case-based model*** of the subject domain can be constructed. The truth value of a sentence in the case-based model is the set of cases for which the sentence is true in a strict sense. From the case-based model fuzzification we obtain a ***fuzzy model***, in which the truth values of the sentences are numbers in the interval [0, 1]. By fuzzifying a set of case-based models, we obtain a ***generalized fuzzy model***. A formal (model-theoretic) description of these models can be found in (Pulchunov and Yakhyaeva, 2005) and (Yakhyaeva, 2007).

Knowledge-based systems are required to exploit knowledge from multiple sources to solve increasingly difficult problems. Therefore there is a need to establish a mechanism of knowledge integration. Many researchers in different subject domains are interested in that problem and consider it from different points of view. Haddad and Bozdogan (2009) provide definition for the knowledge integration phenomenon at both the

Yakhyaeva G. and Yasinkaya O..
An Algorithm to Compare Computer-security Knowledge from Different Sources.
DOI: 10.5220/0005347205650572
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 565-572
ISBN: 978-989-758-096-3

conceptual and operational levels. Steier et al., (1993) implemented knowledge source integration mechanism in Soar architecture system. Console et al., (1991) analyzed integration of different knowledge sources in model-based diagnostic system. Semi-automatic integration of knowledge sources using semantic knowledge articulation tool (SKAT) was provided by Mitra et al., (1999).

One way to generate new knowledge through texts in natural language by comparing and integrating knowledge from different texts (Pulchunov, 2009). While extracting knowledge from natural language texts, different generalized fuzzy models are built. Accordingly, there is a need to ***compare*** the different algebraic systems.

This paper presents a model-theoretic description of comparing knowledge learned from different texts in natural language and an application of this theory in the subject domain of computer security.

2 MATHEMATICAL FOUNDATIONS OF THE DEVELOPED APPROACH

2.1 Case-based Models

First, we define a finite set of documents, each describing some case of computer attacks. We describe each case by algebraic system $\mathfrak{A} = \langle A, \sigma \rangle$, where A is the universe of the algebraic system and σ is its signature. Signature σ is a set of concepts that describe this subject domain: the set of vulnerabilities, threats, countermeasures, consequences, and so on. We assume that all these cases have the same signature. We denote set A and signature σ as $\sigma_A = \sigma \cup \{c_a \mid a \in A\}$. Algebraic systems by which we describe instances of domain belong to the following class

$$\mathbb{K}(\sigma_A) \leftrightharpoons \left\{\mathfrak{A} = \langle \{c_a^{\mathfrak{A}} \mid a \in A\}, \sigma_A \rangle \mid c_a^{\mathfrak{A}} \neq c_b^{\mathfrak{A}} \; if \; a \neq b\right\}.$$

Let $\wp(X)$ denote the set of all subsets of X, and let $S(\sigma_A)$ denote the set of all sentences of the signature σ_A.

The algebraic system $\mathfrak{A}$, a model of some computer attack, will be called the **case** of the considered subject domain. For each set of cases E we define a **case-based model** $\mathfrak{A}_E$.

Definition 1. *Let $E \subseteq \mathbb{K}(\sigma_A)$ be a set of **cases**. A system $\mathfrak{A}_E \leftrightharpoons \langle A, \sigma, \tau_E \rangle$, where $\tau_E : S(\sigma_A) \to \wp(E)$, is called a **case-based model** (generated by the set E) if*

$$\tau_E(\varphi) = \{\mathfrak{A} \in E \mid \mathfrak{A} \vDash \varphi\}$$

for any sentence φ of the signature σ_A.

The case-based model is a Boolean model in which the truth values of the sentences are the elements of Boolean algebra. In this case, the truth values of the sentences are the elements of Boolean algebra of all subsets of set E.

Consider set X to be the set of all kinds of computer attacks: those that have already occurred, and those that can still occur. At any one moment, our knowledge of cyber attacks that have already happened is finite. However, this knowledge is constantly growing, adding new cases. Thus, we can assume that set X can be counted. It is sufficient to consider only the finite subsets of X to formalize our knowledge about the domain at different times. Thus, we consider only finite sets of cases. Denote a class of all finite case-based models

$$\mathbb{K}^f \leftrightharpoons \{\mathfrak{A}_E \mid E \subseteq \mathbb{K}(\sigma_A) \; and \; \|E\| < \omega\}.$$

Suppose we have a case-based model $\mathfrak{A}_E$, a mathematical formalization of the knowledge base of computer-attack cases. To calculate the objective probabilities of different attacks occurring, we define the notion of the **fuzzy model**.

Definition 2. *Let $\mathfrak{A}_E \in \mathbb{K}^f$ be a base-based model. A system $\mathfrak{A}_\mu = \langle A, \sigma_A, \mu \rangle$ is called a **fuzzy model**, generated by the model $\mathfrak{A}_E$ (denoted $\mathfrak{A}_\mu = Fuz(\mathfrak{A}_E)$) if $\mu(\varphi) = \frac{\|\tau_E(\varphi)\|}{\|E\|}$*

for any sentence φ of the signature σ_A.

We introduce the notation for class of all fuzzy models, generated by the models from class $\mathbb{K}^f$

$$\mathbb{K}^\mu \leftrightharpoons \left\{\mathfrak{A}_\mu \mid \exists \mathfrak{A}_E \in \mathbb{K}^f : \mathfrak{A}_\mu = Fuz(\mathfrak{A}_E)\right\}.$$

In practice, one cannot have full information of a considered subject domain. For example, we cannot have information about all the cyber attacks and information-security violations that have occurred. Also, the documentation of particular attacks may be incomplete or inaccurate. It is impossible to give a complete description of the case-based model describing this subject domain, so we must consider fuzzy models that describe the properties of the subject domain that are already known. To formally describe this situation, we introduce the concept of the **generalized fuzzy model**.

Definition 3. *Let $K \subseteq \mathbb{K}^\mu$ and $K \neq \emptyset$. A system $\mathfrak{A}_K = \langle A, \sigma_A, \xi_K \rangle$ is called a **generalized fuzzy model** (generated by the class K) if*

$$\xi_K(\varphi) = \left\{\alpha \in [0,1] \mid \exists \mathfrak{A}_\mu \in K : \mu(\varphi) = \alpha\right\}$$

for any sentence φ of the signature σ_A.

2.2 Principle of Comparing the Generalized Fuzzy Models

One interpretation of the generalized fuzzy model is as follows: Suppose there is some expert in a subject domain described by the language σ_A. For example, this expert may be the system administrator of an enterprise, and the subject domain may be computer security. The expert must deal with a set of situations—the cases of that subject domain—for example, a set of cyber attacks. This set of cases can be considered as a probability space. The cases are the elementary outcomes of this probability space. Naturally, the expert does not know the full description of each case or the truth value of all sentences of signature σ_A for each case. Nevertheless, the expert can estimate the probabilities of the truth values of the sentences based on known information. For example, an expert can claim that 70% of computer attacks use denial-of-service attacks or that not less than 60% of cyber attacks are done to steal information. Previous studies (Pulchunov and Yakhyaeva, 2010) have shown how such probabilistic expert knowledge can be formalized into a generalized fuzzy model.

Now, suppose we have several experts in the subject domain. Each expert has unique knowledge of the subject domain, as they may get their knowledge from different sources and may have different training. When making a decision, we would like to account for the opinions of all the experts to find a compromise.

This problem can be described in formal language as follows: Let the subject domain be described by a signature σ_A, where A is the set of individuals (basic set) taken from the total set of individuals (basic set) in this subject domain. To describe the domain, we construct a finite number of generalized fuzzy models as

$$\{\mathfrak{A}_{K_i} = \langle A, \sigma_A, \xi_i \rangle \mid i = 1, \dots, n\},$$

where K_i is the set of case-based models that generate the model $\mathfrak{A}_{K_i}$, and ξ_i is the evaluation of all sentences of the signature σ_A in model $\mathfrak{A}_{K_i}$. Note that the truth values of a sentence in the generalized fuzzy model are different subsets of the interval [0, 1].

Then, the problem of comparing a finite number of models $\mathfrak{A}_{K_1}, \dots, \mathfrak{A}_{K_n}$ can be formulated as follows: the description of the procedure (algorithm) allowing for any $\varphi \in S(\sigma_A)$ based on the truth values $\xi_1(\varphi), \dots, \xi_n(\varphi)$ of this sentence on models $\mathfrak{A}_{K_1}, \dots, \mathfrak{A}_{K_n}$ build a consistent truth value

$$Tr(\varphi) \subseteq [0, 1].$$

This problem can be formalized by constructing an *n*-ary function

$$f\colon (\rho([0, 1]))^n \to \rho([0, 1]).$$

While constructing this function, consistent truth values for different sentences should not contradict each other. For example, it would be strange for our comparison principle to produce

$$Tr(\varphi) = Tr(\neg\varphi) = 1.$$

Thus, it is more reasonable to formulate the principle of comparing *n* generalized fuzzy models as an *n*-ary function f defined by the set of all generalized fuzzy models:

$$f\colon \langle \mathfrak{A}_{K_1}, \dots, \mathfrak{A}_{K_n} \rangle \mapsto \mathfrak{A}_K .$$

Moreover, it would be ideal for this comparison principle to work on any finite set of models and to not depend on the order the models are considered. These properties are achieved by using the properties of associativity and commutativity.

2.3 Product of the Generalized Fuzzy Models

First we define the operation of the product on the class $\mathbb{K}^f$ of case-based models.

Definition 4. *Let* $\mathfrak{A}_{E_1}, \mathfrak{A}_{E_2}$ *be case-based models. We assume that* $E_1 \cap E_2 = \emptyset$ *(perhaps, after renaming). A model* $\mathfrak{A}_E$ *is called the* ***product*** *of* $\mathfrak{A}_{E_1}$ *and* $\mathfrak{A}_{E_2}$*, denoted as* $\mathfrak{A}_E = \mathfrak{A}_{E_1} * \mathfrak{A}_{E_2}$*, if:*

1) $E = E_1 \cup E_2$;

2) $\tau_E(\varphi) = \tau_{E_1}(\varphi) \cup \tau_{E_2}(\varphi)$ *for any* $\varphi \in S(\sigma_A)$.

Paper (Pulchunov and Yakhyaeva, 2010) proved that the operation * is associative, commutative, and closed in a set of case-based models.

Statement 5. *Let* $\mathfrak{A}_E = \mathfrak{A}_{E_1} * \mathfrak{A}_{E_2}$ *and* $\mathfrak{A}_{\mu_1} = Fuz(\mathfrak{A}_{E_1}), \mathfrak{A}_{\mu_2} = Fuz(\mathfrak{A}_{E_2}), \mathfrak{A}_{\mu} = Fuz(\mathfrak{A}_E)$. *Then,*

$$\mu(\varphi) = \frac{\mu_1(\varphi) \cdot \|E_1\| + \mu_2(\varphi) \cdot \|E_2\|}{\|E_1\| + \|E_2\|}$$

for any $\varphi \in S(\sigma_A)$.

A proof of this statement can also be found in (Pulchunov and Yakhyaeva, 2010).

Consequence 6. *Let* $\mathfrak{A}_E = \mathfrak{A}_{E_1} * \mathfrak{A}_{E_2}$ *and* $\mathfrak{A}_{\mu_1} = Fuz(\mathfrak{A}_{E_1}), \mathfrak{A}_{\mu_2} = Fuz(\mathfrak{A}_{E_2}), \mathfrak{A}_{\mu} = Fuz(\mathfrak{A}_E)$. *Then,*

$min\{\mu_1(\varphi), \mu_2(\varphi)\} \le \mu(\varphi) \le max\{\mu_1(\varphi), \mu_2(\varphi)\}$ *for any* $\varphi \in S(\sigma_A)$.

Now we can define the operation of the product on a set of generalized fuzzy models.

Definition 7. *Consider the generalized fuzzy models* $\mathfrak{A}_{K_1}$ *and* $\mathfrak{A}_{K_2}$. *Let* $K_1 * K_2 =$

$$\{Fuz(\mathfrak{A}_{E_1} * \mathfrak{A}_{E_2}) | Fuz(\mathfrak{A}_{E_1}) \in K_1, Fuz(\mathfrak{A}_{E_2}) \in K_2\}.$$

Then, the generalized fuzzy model $\mathfrak{A}_{K_1 * K_2}$ *is called the* ***product*** *of models* $\mathfrak{A}_{K_1}$ *and* $\mathfrak{A}_{K_2}$.

Because the product of case-based models is commutative and associative, the product of the generalized fuzzy models will also be commutative and associative.

3 COMPUTER SECURITY SOFTWARE

3.1 Knowledge Base

We developed a software system called RiskPanel, essentially a workplace for experts to ensure the security of corporate information, based on the methodology of generalized fuzzy models (Pulchunov et al., 2011).

The core of this system is an information-security knowledge base. To organize and work with the knowledge base, we use OntoBox technology (Malykh and Mantsivoda, 2010). This system represents and stores data in an ontological format and has powerful, flexible processing tools. It allows for great modularity and portability of knowledge bases, advantageous when developing complex information systems.

Seven categories of attributes (classes) were created to describe the cases in the OntoBox knowledge base: symptoms, threats, vulnerabilities, consequences, loss, countermeasures, and configurations. Each attribute category was represented with a tree structure. The cases in the database are characterized by certain attributes of each category. Each case is formed based on natural-language text found on the Internet (Yakhyaeva and Yasinskaya, 2012).

While analyzing the texts provided to form the cases, we found most of them had clear but not full information. In other words, we could not perfectly describe whether a particular case had specific knowledge-base attributes. To solve this problem, we proposed using an open-world semantic methodology, widely used in description logic systems (Baader, 2003). Basically, this approach considers all possible interpretations of unknown information. Thus, to mathematically describe a computer-attack case, we consider a generalized fuzzy model with certain attributes, called a partial case.

Definition 8. *Consider a set* $U \subseteq S(\sigma_A)$ *and evaluation* $\nu: U \to \{0, 1\}$. *We say that Case* $\mathfrak{A}$ *is* ***consistent*** *with the evaluation* ν *(and denote* $\mathfrak{A} \uparrow \nu$*) if* $\mathfrak{A} \vDash \varphi \iff \nu(\varphi) = 1$ *for any* $\varphi \in U$.

Definition 9. *Consider a set* $U \subseteq S(\sigma_A)$ *and evaluation* $\nu: U \to \{0, 1\}$. *A generalized fuzzy model* $\mathfrak{A}_K$ *is called a* ***generalized case*** *(generated by the evaluation* ν*) if*

$$K = \{\mathfrak{A} \mid \mathfrak{A} \in K(\sigma_A) \text{ and } \mathfrak{A} \uparrow \nu\}.$$

In this formalism, the entire knowledge base of RiskPanel can be considered a finite set of generalized cases. When drawing conclusions from this knowledge base, we must compare these models.

For a knowledge base formalized as a set of generalized cases, it is most appropriate to use a comparison principle based on the product of the generalized fuzzy models, because it is consistent with open-world semantics.

Note that each generalized case $\mathfrak{A}_K$ is not an interval model. Moreover, for each sentence $\varphi \in S(\sigma_A)$, the truth value $\xi_K(\varphi)$ belongs to $\{\{0\}, \{1\}, \{0, 1\}\}$. Now, we must formulate an algorithm for calculating the truth values in a consistent model of generalized cases.

Theorem 10. *Let* $\mathfrak{A}_{K_1}, \dots, \mathfrak{A}_{K_n}$ *be generalized cases. Then, for* $\varphi \in S(\sigma_A)$ *we have*

$$\xi_{K_1 * \dots * K_n}(\varphi) = \left\{\frac{\alpha}{n}; \frac{\alpha + 1}{n}; \dots; \frac{\alpha + \beta}{n}\right\},$$

where

$$\alpha = \left\|\{\mathfrak{A}_{K_i} \mid \xi_{K_i}(\varphi) = \{1\}\}\right\|$$

and

$$\beta = \left\|\{\mathfrak{A}_{K_i} \mid \xi_{K_i}(\varphi) = \{0, 1\}\}\right\|.$$

Proof. Let $q \in \xi_{K_1 * \dots * K_n}(\varphi)$. Then, there are such $\mathfrak{A}_1 \in K_1, \dots, \mathfrak{A}_n \in K_n$ such that (see Statement 5)

$$q = \frac{\varepsilon_1(\varphi) + \dots + \varepsilon_n(\varphi)}{n},$$

where for any $i = 1, \dots, n$ if $\mathfrak{A}_i \vDash \varphi$ then we have $\varepsilon_i(\varphi) = 1$, and if $\varepsilon_i(\varphi) = 0$ then we have $\mathfrak{A}_i \nvDash \varphi$. Obviously,

$$\alpha \le \varepsilon_1(\varphi) + \dots + \varepsilon_n(\varphi) \le \alpha + \beta.$$

Thus, we obtain $q \in \left\{\frac{\alpha}{n}; \frac{\alpha+1}{n}; \dots; \frac{\alpha+\beta}{n}\right\}$.

Now consider $q \in \left\{\frac{\alpha}{n}; \frac{\alpha+1}{n}; \dots; \frac{\alpha+\beta}{n}\right\}$. Let $q = \frac{\gamma}{n}$. Obviously, $\alpha \le \gamma \le \alpha + \beta$. Let

$$A = \{\mathfrak{A}_{K_i} \mid \xi_{K_i}(\varphi) = \{1\}\} = \left\{\mathfrak{A}_{K_{i_1}}, \dots, \mathfrak{A}_{K_{i_\alpha}}\right\},$$

$$B = \left\{\mathfrak{A}_{K_j} \mid \xi_{K_j}(\varphi) = \{0,1\}\right\} = \left\{\mathfrak{A}_{K_{j_1}}, \dots, \mathfrak{A}_{K_{j_\beta}}\right\}.$$

First, we select one case from each generalized case of set A, denoting them as $\mathfrak{A}_{i_s} \in K_{i_s}$ $(s = 1, \dots, \alpha)$. Then we select cases $\mathfrak{A}_{j_s} \in K_{j_s}$ $(s = 1, \dots, \gamma - \alpha)$ from the generalized cases of set B such that $\mathfrak{A}_{j_s} \vDash \varphi$. Last, we select cases $\mathfrak{A}_{j_s} \in K_{j_s}$ $(s = \gamma - \alpha + 1, \dots, \beta)$ from the generalized cases of set B such that $\mathfrak{A}_{j_s} \nvDash \varphi$.

Obviously,

$$\mathfrak{A}_{i_1} * \dots * \mathfrak{A}_{i_\alpha} * \mathfrak{A}_{j_1} * \dots * \mathfrak{A}_{j_\beta} \vDash_q \varphi.$$

Thus, $q \in \xi_{K_1 * \dots * K_n}(\varphi)$.

Note that comparing the finite set of generalized cases will not produce an interval model. But, when $n \to \infty$, the truth values of sentences in a consistent model will tend toward intervals on the set $[0, \dots, 1] \cap \mathbb{Q}$. Thus, in practice, when dealing with a large enough set of cases, we can view the truth values in a consistent model as intervals.

3.2 Theorem of Atomically Generalized Cases

Definition 11. A *generalized case is called an* ***atomically generalized case*** *if it is generated by evaluating the subset of the set of all atomic propositions.*

Consider a quantifier-free sentence $\varphi(A_1, \dots, A_n)$ from n atomic propositions. Let us reduce this sentence to PDNF:

$$\varphi(A_1, \dots, A_n) = \omega_1 \vee \dots \vee \omega_k,$$

where ω_i $(i \in \{1, \dots, k\})$ are the elementary conjunctions consisting of atomic propositions $A_1, \dots, A_n$.

We introduce the following notation:

$$Con(\varphi) = \{\omega_1, \dots, \omega_k\},$$

$$Con(\varphi, \{0\}) = \{\omega \in Con(\varphi) \mid \xi_K(\omega) = \{0\}\},$$

$$Con(\varphi, \{1\}) = \{\omega \in Con(\varphi) \mid \xi_K(\omega) = \{1\}\},$$

$$Con(\varphi, \{0,1\}) = \{\omega \in Con(\varphi) \mid \xi_K(\omega) = \{0,1\}\}.$$

Theorem 12. *Let* $\mathfrak{A}_K$ *be an atomically generalized case, and let* φ *be a quantifier-free sentence of signature* σ_A*. Then,*

$$\xi_K(\varphi) = \begin{cases} \{0\}, & Con(\varphi) = Con(\varphi, \{0\}); \\ \{1\}, & (Con(\varphi, \{1\}) \neq \emptyset)\ or \\ & \left(\|Con(\varphi, \{0,1\})\| = 2^{\|(A_i \mid \xi_K(A_i) = \{0,1\})\|}\right); \\ \{0,1\}, & otherwise. \end{cases}$$

Proof. Obviously,

$$\xi_K(\varphi) = \{0\} \Leftrightarrow \forall \mathfrak{A} \in K\ (\mathfrak{A} \nvDash \varphi) \Leftrightarrow$$
$$\Leftrightarrow \forall \mathfrak{A} \in K\ (\mathfrak{A} \nvDash \omega_1, \dots \mathfrak{A} \nvDash \omega_n) \Leftrightarrow$$
$$\Leftrightarrow \xi_K(\omega_1) = \dots = \xi_K(\omega_n) = \{0\} \Leftrightarrow$$
$$\Leftrightarrow Con(\varphi) = Con(\varphi, \{0\}).$$

On the other hand,

$$Con(\varphi, \{1\}) \neq \emptyset \Leftrightarrow \exists \omega_i\, \forall \mathfrak{A} \in K\ (\mathfrak{A} \vDash \omega_i) \Rightarrow$$
$$\Rightarrow \forall \mathfrak{A} \in K\ (\mathfrak{A} \vDash \varphi) \Leftrightarrow \xi_K(\varphi) = \{1\}.$$

Let $\{A_1, \dots, A_n\}$ be a set of atomic propositions included in φ. Consider the set of elementary conjunctions

$$V = \left\{A_1^{\varepsilon_1} \& \dots \& A_n^{\varepsilon_n} \mid \exists \mathfrak{A} \in K: \mathfrak{A} \vDash A_1^{\varepsilon_1} \& \dots \& A_n^{\varepsilon_n}\right\}.$$

Let $\alpha = \|(A_i \mid \xi_K(A_i) = \{0,1\})\|$. Obviously, $\alpha \neq 0$; otherwise, $Con(\varphi, \{0,1\}) = \emptyset$. Consequently, $\|V\| = 2^\alpha$.

Assume now that $Con(\varphi) \neq Con(\varphi, \{0\})$ and $Con(\varphi, \{1\}) = \emptyset$. Thus, $Con(\varphi, \{0,1\}) \neq \emptyset$. Moreover, $Con(\varphi, \{0,1\}) \subseteq V$.

Consider two cases: $Con(\varphi, \{0,1\}) = V$ and $Con(\varphi, \{0,1\}) \neq V$.

Let $Con(\varphi, \{0,1\}) = V$. Then, for any case $\mathfrak{A} \in K$, there is a conjunct $\omega_i \in Con(\varphi, \{0,1\})$ such that $\mathfrak{A} \vDash \omega_i$. Consequently, $\xi_K(\varphi) = \{1\}$ for any case $\mathfrak{A} \in K$.

Assume now that $Con(\varphi, \{0,1\}) \subset V$. Then there is a case $\mathfrak{A}' \in K$ such that $\mathfrak{A}' \nvDash \omega_i$ for any $\omega_i \in Con(\varphi, \{0,1\})$. Because we have assumed that $Con(\varphi, \{1\}) = \emptyset$, then $\mathfrak{A}' \nvDash \varphi$. On the other hand, because $Con(\varphi, \{0,1\}) \neq \emptyset$, there is a case $\mathfrak{A}''$such that $\mathfrak{A}'' \vDash \varphi$. Thus, $\xi_K(\varphi) = \{0,1\}$.

3.3 Module of Knowledge Comparison

RiskPanel has a module for comparing knowledge learned from various computer-attack cases. Currently, the module interface allows one to calculate the truth value as an interval for a formula presented in PDNF (perfect disjunctive normal form).

Consider the module interface (Fig. 1). To input data into the main algorithm, the user must enter the

parameters of the formula using the resources provided. First, the user must select the attributes included in all conjunctions of PDNF. Next, the user must specify the number of conjunctions in the formula. Then, drop-down lists of «+» and «–» values appear with the resulting PDNF, where «–» symbolizes negation of the argument. The data from this window with PDNF can be inputted into the main algorithm by clicking the button titled «Get the value of the formula».

The value of the formula is calculated as an interval (see Theorem 10). The start value of the interval is the ratio of the number of cases for which the formula is true to the number of all existing cases. The end value of the interval is the ratio of the number of cases for which the formula is true, added to the number of cases for which the truth value of the formula is not defined, to the number of all existing cases.

The algorithm used to determine the truth value of a formula in the generalized case is based on Theorem 12 and shown in Table 1. At first, false conjunctions that contradict the available information for the case are eliminated from the formula. If no conjunctions in the formula remain, then the formula for the case is false. If the remaining conjunctions do not have unknown attribute values for the case, then the formula for the case is considered true. If the remaining conjunctions have unknown attribute values, then the algorithm operates as follows: If the number of remaining conjunctions is less than 2^n, where n is the number of unknown attribute values in the conjunction, then the truth value of the formula for the case is not defined, otherwise the formula is true.

To determine whether a case has attribute values included in the conjunctions requires $O(n)$ operations, where n is the number of attribute values in all categories stored in OntoBox. To eliminate false conjunctions for the case based on the information of attribute values requires $O(k)$ operations, where k is the number of conjunctions in PDNF. The total number of attribute values involved in the conjunctions cannot exceed n. Thus, the total algorithmic complexity of the developed approach for defining the truth value of PDNF in a case is $O(n(n+k))$.

Further, if the OntoBox knowledge base has m computer-attack cases, then calculating the truth value of the formula in interval form needs $O(mn(n+k))$ operations.

Table 1: The algorithm for determining the truth value of a formula in the generalized case.

```
alg getPDNFVerityOnCase(arg Case case,
arg list PDNFFormulaAttrs,
arg matrix PDNFBoolMatrix)
begin
| bool rightValue,
| int unknownAttrsCount,
| list removedIndexes
| for each Attribute attr in
| |              PDNFFormulaAttrs
| | int attrValueOnCase :=
| |     checkIfCaseHasAttr(attr, case)
| | if (attrValueOnCase = UNKNOWN_ATTR)
| | | unknownAttrsCount :=
| | |             unknownAttrsCount + 1
| | else
| | | if (attrValueOnCase = HAS_ATTR)
| | | | rightValue := true
| | | |   else
| | | | rightValue := false
| | | list boolRow :=
| | | |   PDNFBoolMatrix.get(
| | | |   PDNFFormulaAttrs.indexOf(attr))
| | | for int i = 0 to boolRow.size()
| | | | if (removedIndexes does not
| | | | |                    contain i)
| | | | | if (boolRow.get(i) !=
| | | | | |                  rightValue)
| | | | | | removedIndexes.add(i)
| | | |   end of loop
| | end of loop
| int remainingConjCount :=
|        PDNFBoolMatrix.get(0).size() -
|        removedIndexes.size()
| if (remainingConjCount = 0)
| | return PDNF_FALSE
| if (unknownAttrsCount = 0)
| | return PDNF_TRUE
| if (remainingConjCount <
| |      2^unknownAttrsCount)
| | return PDNF_UNKNOWN
| else
| | return PDNF_TRUE
end
```

4 CONCLUSIONS

This work describes the mathematical apparatus and software implementation of one of the modules of the RiskPanel system, aimed to compare computer-security knowledge learned from various online sources.

Algorithms implemented in this module are based on the methodology of generalized fuzzy models. The knowledge obtained from a single source is formalized as an algebraic system and is stored in the knowledge base of the RiskPanel system. To implement the knowledge comparison,

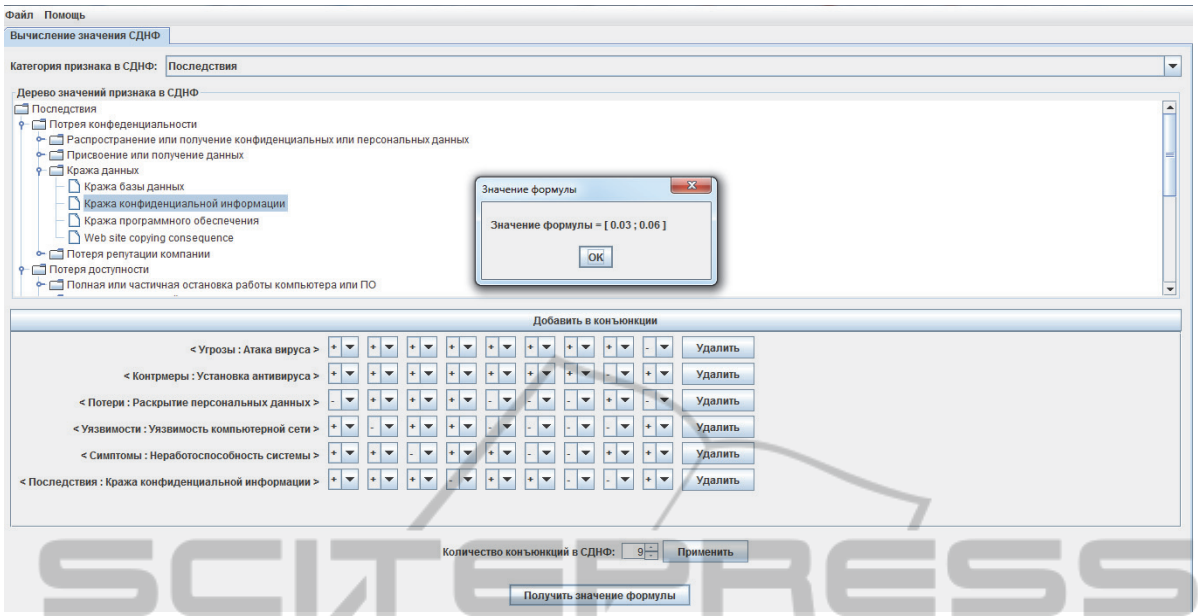

Figure 1: Module of Knowledge Comparison.

we construct a generalized fuzzy model, the product of all algebraic systems stored in the database.

The system interface allows one to calculate the truth value of any quantifier-free sentence. The input sentence is presented in PDNF. The truth value is calculated as a probability interval.

The developed algorithm has polynomial complexity.

ACKNOWLEDGEMENTS

The research for this paper was financially supported by the Ministry of Education of the Russian Federation (project no. 2014/139) and was partially supported by RFBR (project no. 14-07-00903-a).

REFERENCES

Assali, A., Lenne, D. & Debray, B., 2013. Adaptation Knowledge Acquistion in a CBR System. *International Journal on Artificial Intelligence Tools,* 22(1).

Baader, F., 2003. *The Description Logic Handbook.* Ney York: Cambridge University Press.

Burger, J. et al., 2013. Model-Based Security Engineering: Managed Co-evolution of Security Knowledge and Software Models. *Foundation of Security Analysis and Design VII - FOSAD 2012/2013 Tutorial Lectures. Springer Lecture Notes in Computer Science,* pp. 34-53.

Console, L., Theseider, D. & Torasso, P., 1991. Towards the integration of different knowledge sources in model-based diagnosis. *Trends in Artifician Intelligence, Lecture Notes in Computer Science,* Volume 549, pp. 177-186.

Gartner, S. et al., 2014. Maintaining requirements for long-living software systems by incorporating security knowledge. *IEEE 22nd International Requirements Engineering Conference,* pp. 103-112.

Haddad, M. & Bozdogan, K., 2009. *Knowledge Integration in Large-Scale Organizations and Networks - Conceptual Overviev and Operational Definition.* [Online] Available at: http://dx.doi.org/10.2139/ssrn.1437029

Kolodner, J., 1992. An introduction to Case-based reasoning. *Artificial Intelligence Review,* Volume 6, pp. 3-34.

Malykh, A. & Mantsivoda, A., 2010. Query Language for Logic Architectures. *Perspectives of System Informatics: Proceedings of 7th International Conference. Lecture Notes in Computer Science*, Volume 5947, pp. 294-305.

Mitra, P., Wiederhold, G. & Jannink, J., 1999. *Semi-automatic Integration of Knowledge Sources.* Sunnyvale, CA, July 6-8, 2-nd International Conference on Information Fusion.

Palchunov, D., 2008. The solution of the problem of information retrieval based on ontologies. *Bisnes-informatika,* 1(1), pp. 3-13.

Pulchunov, D., 2009. Knowledge search and production: creation of new knowledge on the basis of natural language text analysis. *Filosofiya nayki,* 43(4), pp. 70-90.

Pulchunov, D. & Yakhyaeva, G., 2005. Interval fuzzy algebraic systems. *Proceedings of the Asian Logic*

Conference , pp. 23-37.

Pulchunov, D. & Yakhyaeva, G., 2010. Fuzzy algebraic systems. *Vestnik NGU. Seriya: Matematica, mexanika, informatika,* 10(3), pp. 75-92.

Pulchunov, D., Yakhyaeva, G. & Hamutskya, A., 2011. Software system for information risk manadgement "RiskPanel". *Programmnaya ingeneriya,* Volume 7, pp. 29-36.

Ruhroth, T. et al., 2014. Towards Adaptation and Evolution of Domain-Specific Knowledge for Maintaining Secure Systems. *15th International Conference on Product-Focused Software Process Improvement, Springer Lecture Notes in Computer Science,* pp. 239-253.

Steier, D., Lewis, R., Lehman, J. & Zacherl, A., 1993. Combining multiple knowledge sources in an integrated intelligent system. *IEEE Expert,* 8(3), pp. 35-44.

Thayse, F., 1989. *From Modal Logic to Deductive Databases: Introduction a Logic Based Approach to Artificial Intelligence.* Chichester: Wiley.

Yakhyaeva, G., 2007. *Fuzzy model truth values.* Bratislava, Proceedings of the 6-th International Conference Aplimat, pp. 423-431.

Yakhyaeva, G. & Yasinskaya, O., 2012. The application of precedent model methodology in the risk-management system aimed at early detection of computer attacks. *Vestnik NGU. Seriya: Informationnie Texnologii,* 10(2), pp. 106-115.

Yakhyaeva, G. & Yasinskaya, O., 2014. Application of Case-based Methodology for Early Diagnosis of Computer Attacks. *Journal of Computing and Information Technology*, 22(3), p. 145–150.