**IEEE**Access

Multidisciplinary : Rapid Review : Open Access Journal

# A Method for Knowledge Representation to Design Intelligent Problems Solver in Mathematics Based on Rela-Ops Model

**HIEN D. NGUYEN**[1,2], **NHON V. DO**[3], **VUONG T. PHAM**[4], **ALI SELAMAT**[5,6,7,8], **(Member, IEEE),**
**AND ENRIQUE HERRERA-VIEDMA**[9,10], **(Senior Member, IEEE)**

[1]Faculty of Computer Science, University of Information Technology, Ho Chi Minh, Vietnam
[2]Vietnam National University, Ho Chi Minh 700000, Vietnam
[3]Artificial Intelligence Department, Faculty of Engineering and Technology, Hong Bang International University, Ho Chi Minh 700000, Vietnam
[4]Faculty of Information Techonology, Sai Gon University, Ho Chi Minh 700000, Vietnam
[5]Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia
[6]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia
[7]Media and Games Center of Excellence (MagicX), Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia
[8]Faculty of Informatics and Management, University of Hradec Kralove, 50003 Hradec Kralove, Czech Republic
[9]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[10]Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Ali Selamat (aselamat@utm.my)

**ABSTRACT** Knowledge-base is a fundamental platform in the architecture of an intelligent system. Relations and operators are popular knowledge in practice knowledge domains. In this paper, we propose a method to represent the model by combining these kinds of knowledge, called the Rela-Ops model. This model includes foundation components consisting of concepts, relations, operators, and inference rules. It is built based on ontology and object-oriented approaches. Besides the structure, each concept of the Rela-Ops model is a class of objects which also have behaviors to solve problems on their own. The processing of algorithms for solving problems on the Rela-Ops model combines the knowledge of relations and operators in the reasoning. Furthermore, we also propose a knowledge model for multiple knowledge domains, in which each sub-domain has the form as the Rela-Ops model. These representation methods have been applied to build knowledge bases of Intelligent Problems Solver (IPS) in mathematics. The knowledge base of 2D-Analytical Geometry in a high-school is built by using the Rela-Ops model, and the knowledge base of Linear Algebra in university is designed by using the model for multiple knowledge domains. The IPS system can automatically solve basic and advanced exercises in respective courses. The reasoning of their solutions is done in a step-by-step approach. It is similar to the solving method by humans. The solutions are also pedagogical and suitable for the learner's level and easy to be used by students studying 2D-Analytical Geometry in high-school and Linear Algebra in university.

**INDEX TERMS** Knowledge representation, knowledge-based systems, intelligent problem-solver, knowledge engineering.

## I. INTRODUCTION

Science Technology Engineering and Math (STEM) is a modern approach for studying in the technological era [3]. Building the Intelligent Problem Solver (IPS) in STEM, especially for mathematical courses, is a grand challenge for artificial intelligence in education [1]–[3]. This system can automatically solve problems. Those problems are declared

their hypothesis and goal by using a specification language [4], [5]. The system will automatically solve them or give some instructions to solve them. Besides the ability to solve common exercises, this system requires the pedagogy. Their proofs are suitable for the learner's level. They also help the learner understanding the method for solving problems. An IPS in education can solve many kinds of exercises from basic to advanced kinds in the course. It has to satisfy these requirements [4], [6], [36], [48]:

- The program has an adequate and useful knowledge base.

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

- The program can solve conventional kinds of exercises in the course.
- The program is pedagogical.
- The program is useful for studying.

For meeting these requirements, according to [4], [6], [7], [48], a method for building the knowledge base of IPS has to satisfy the following criteria:

*Universality [4], [6], [48]:* This criterion provides the flexibility of the representation method. The method can be applied to representing various knowledge domains [4], [6]. It can be used directly or improved with some little things to use. A model has to represent the foundation of the knowledge domain, including concepts, relations between concepts, and inference rules [48]. That foundation is a knowledge kernel as ontology. The kernel can integrate with other knowledge to strengthen the ability to represent practical knowledge.

*Usability [4], [7]:* It has the simple specification language being adequately to represent knowledge domains. This language is also easy to employ for representing and updating knowledge domains. When it is applied, the knowledge base of the system gives a natural inference that can be understood by the user.

*Practicality [6], [7]:* The model must be applied to represent various real knowledge domains. This representation is similar to that of humans, and it can solve some practical problems in the knowledge domain.

*Formality [7], [8]:* The structure of the model is built based on the solid mathematical foundation. It is also used to construct the model of problems. The finiteness and effectiveness of algorithms for solving the problems are proven. The complexity of them is also evaluated.

Each criterion has four levels from $1 - 4$, respectively, from very weak – very strong as [6]. The meaning of each level is described in Fig. 1 as follows:

Knowledge of relations and operators is a prevalent form of human knowledge. The model for this knowledge domain consists of concepts, relations, operators, and inference rules. Some methods for representing this knowledge have been studied, such as graph, logic, or ontology. Based on the mathematical structures about relations and operators, some models for representing the knowledge of relations [9], [11] and the knowledge of operators [10], [12] are studied. However, they cannot represent the knowledge domains that have both relations and operators. The result in [13] is using the syntax-semantics model to extract relations between the entities and the geometric attributes pattern; nonetheless, this method is not universality. It is based on the characteristic of geometric knowledge. Although the algebraic structures have also been used to represent this kind of knowledge, those methods have limitations and cannot satisfy the requirements of knowledge representation in intelligent systems [27], [28].

In this paper, a model for combining the knowledge of relations and operators, called the Rela-Ops model, is proposed. This model is built based on ontology and object-oriented approaches. Its foundation includes components: concepts, relations, operators, and inference rules, in which each concept in the Rela-Ops model is a class of objects. Based on its structure, this model also defines the sentence and problems on it. This model is useful in practical applications. In the inference processing, it combines the knowledge of relations and operators to solve current problems in the knowledge domain. Besides, the Rela-Ops model satisfies the criteria of a method for representing the knowledge base of IPS in education. More than that, we present a method to represent a knowledge domain, including multiple sub-domains; each sub-domain has the Rela-Ops model form. These proposed methods have been applied to build knowledge bases of IPS in mathematics. The knowledge base of 2D-Analytical Geometry in a high-school is built by using the Rela-Ops model, and the knowledge base of Linear Algebra in university is designed by using the model for multiple knowledge domains. These systems can solve the usual exercises of respective courses automatically. The reasoning of their solutions is clearly. It is similar to the solving method of humans. Solutions are also pedagogical and suitable for the learner's level.

The next section presents related works about methods constructing the knowledge base of IPS. Section 3 presents the knowledge model of relations and operators, which is called the Rela-Ops model. The syntax of the clauses and predicates on this model are also presented in that section. Section 4 presents problems with the Rela-Ops model and their models. Algorithms for solving them are designed based on their models and knowledge in the Rela-Ops model. Section 5 presents the knowledge model for multiple knowledge domains. Section 6 shows the applications of the proposed methods to build IPS for Linear Algebra in university and 2D-Analytical Geometry in high school. The last section concludes the paper.

## II. RELATED WORK

Nowadays, there are many methods for representing knowledge, including formal logic, frames, networks, ontology, and algebraic approaches. However, they have some limitations to apply in practice, especially in designing knowledge bases of IPS in education.

*Formal logic methods* only can represent simple knowledge domains — many kinds of logic methods being studied for knowledge representation — for example, predicate methods, first-order logic, temporal logic, and description logic [14]–[16]. Logical methods are the formal representation of semantic [14]. In [15], [16], the authors represent the operators and relations in logic by matrices and tensors. Besides that, the study in [44] presents a logical method to describe the semantic and syntactical aspects of uncertain decision implications. This work is built based on a solid mathematical foundation, but those results are theoretical. The specification of inference rules in the real knowledge domain is not suitable for ordinary users, especially the students.
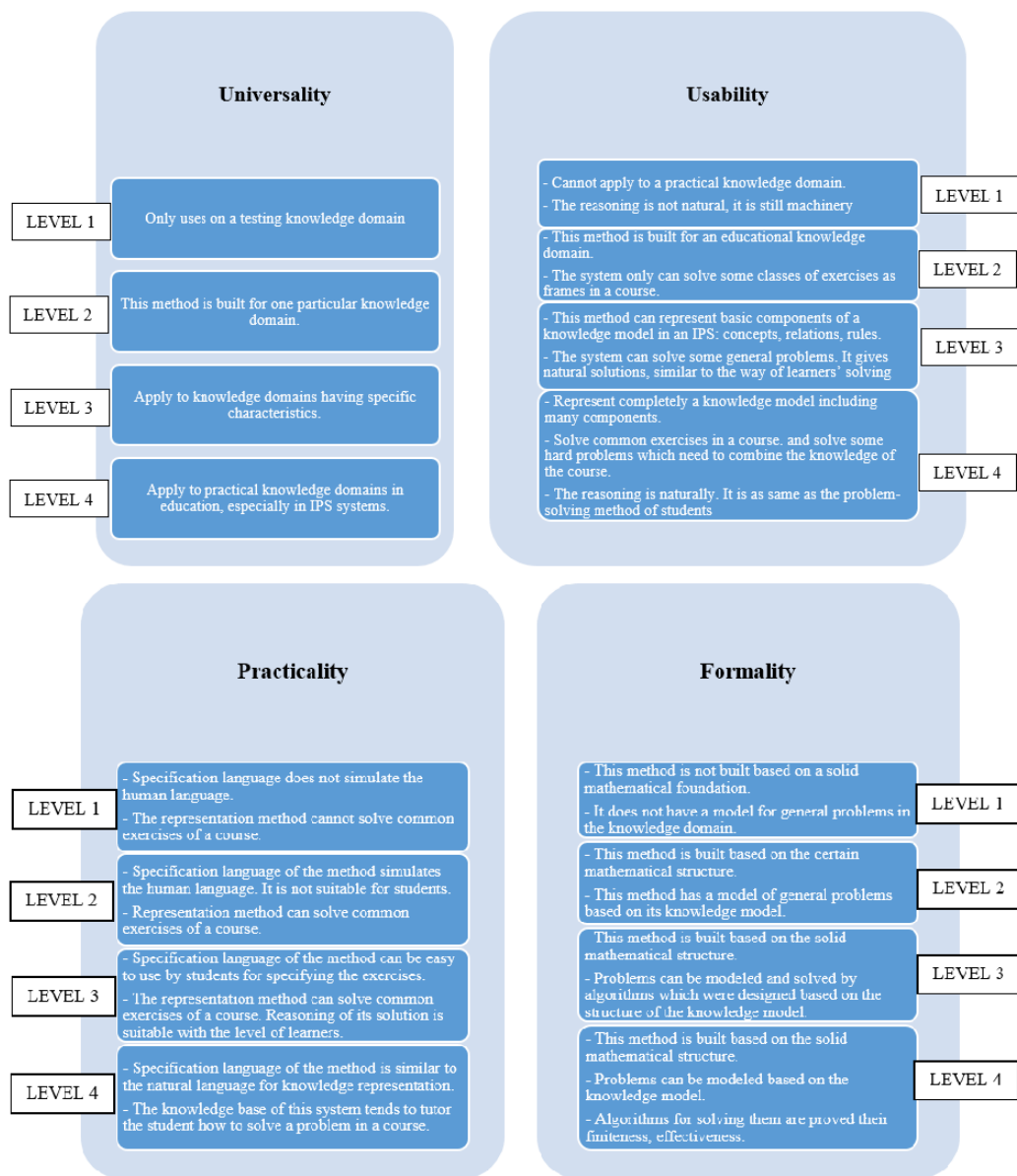
## Universality

| | |
|---|---|
| LEVEL 1 | Only uses on a testing knowledge domain |
| LEVEL 2 | This method is built for one particular knowledge domain. |
| LEVEL 3 | Apply to knowledge domains having specific characteristics. |
| LEVEL 4 | Apply to practical knowledge domains in education, especially in IPS systems. |

## Usability

| | |
|---|---|
| - Cannot apply to a practical knowledge domain. <br> - The reasoning is not natural, it is still machinery | LEVEL 1 |
| - This method is built for an educational knowledge domain. <br> - The system only can solve some classes of exercises as frames in a course. | LEVEL 2 |
| - This method can represent basic components of a knowledge model in an IPS: concepts, relations, rules. <br> - The system can solve some general problems. It gives natural solutions, similar to the way of learners' solving | LEVEL 3 |
| - Represent completely a knowledge model including many components. <br> - Solve common exercises in a course. and solve some hard problems which need to combine the knowledge of the course. <br> - The reasoning is naturally. It is as same as the problem-solving method of students | LEVEL 4 |

## Practicality

| | |
|---|---|
| LEVEL 1 | - Specification language does not simulate the human language. <br> - The representation method cannot solve common exercises of a course. |
| LEVEL 2 | - Specification language of the method simulates the human language. It is not suitable for students. <br> - Representation method can solve common exercises of a course. |
| LEVEL 3 | - Specification language of the method can be easy to use by students for specifying the exercises. <br> - The representation method can solve common exercises of a course. Reasoning of its solution is suitable with the level of learners. |
| LEVEL 4 | - Specification language of the method is similar to the natural language for knowledge representation. <br> - The knowledge base of this system tends to tutor the student how to solve a problem in a course. |

## Formality

| | |
|---|---|
| - This method is not built based on a solid mathematical foundation. <br> - It does not have a model for general problems in the knowledge domain. | LEVEL 1 |
| - This method is built based on the certain mathematical structure. <br> - This method has a model of general problems based on its knowledge model. | LEVEL 2 |
| - This method is built based on the solid mathematical structure. <br> - Problems can be modeled and solved by algorithms which were designed based on the structure of the knowledge model. | LEVEL 3 |
| - This method is built based on the solid mathematical structure. <br> - Problems can be modeled based on the knowledge model. <br> - Algorithms for solving them are proved their finiteness, effectiveness. | LEVEL 4 |

**FIGURE 1.** Levels of each criterion of a method for building a knowledge base of IPS in education.

When applying formal logic methods in knowledge domains about education, the representing is not natural; it is not similarly the way humans representing the knowledge of courses. Hence, those methods are not effective in designing the knowledge base of the system supporting learning.

*Networks* are suitable methods to represent concepts and their relationships in the knowledge domain [17]. However, they are not effective in representing the knowledge domain about computing. *The knowledge graph* performs a collection of interlinked descriptions of entities. The nodes of this graph represent items, and relations of nodes that interact with each other are represented by its edges [18]. This graph is a useful tool for semantic searching and describing the semantics of information. In [19], a knowledge graph embedding with the concepts model is proposed. This model represents the relationships between entities and their concepts. It can adjust a knowledge graph by the concept of information of entities from a concept graph. Besides that, there are some methods to extract relations from multiple knowledge graphs by considering the alignments between them [20]. Nonetheless, the knowledge graph is not sufficient in solving problems by reasoning methods, especially problems of an IPS in education.

Petri nets, which can be used to handle many problems [51], [52], are a modeling technique to construct knowledge-based systems in many fields [45]. They are useful for designing knowledge bases fuzzy reasoning of uncertain expert systems. The model combining Petri nets and information theory is sufficient to consider the development

of a discrete event system. In [46], the authors propose a cloud reasoning Petri nets (CRPN) model to represent an uncertain knowledge domain. It used the operator of interval cloud hybrid averaging. However, the knowledge of mathematical courses cannot be represented by Petri nets, so this model cannot use to design the knowledge bases of IPS in mathematics.

*Frames* are a large part of knowledge representation schemes [21]. Some of IPS systems have been designed based on their knowledge bases as frames, such as symbolab [39], mathway [40]. By using frames, the systems only solve some kinds of problems which were set up, they cannot solve other kinds of problems, especially problems require the use of the depth knowledge or the combining knowledge. In [22], the authors proposed a sharing framework and algorithms for solving problems about explicit arithmetic and proving plane geometry theorems. They used the syntax-semantics model [13], [23] to extract relations for a math problem. That problem was solved through the set of extracted relations. However, this method does not support the learning; it does not show a pedagogical solution which has the reasoning similar to humans.

In intelligent tutoring systems (ITS), the content of a course is represented by *ontology* as a framework [24]. In robotics, ontology represents the knowledge about the environment, events, and actions, which make robots more autonomous, especially in automatic tutoring systems [25], [50]. Ontology COKB (Computational Object Knowledge Base) is an emerging method to build an IPS in education [26]. The mathematical foundation of the COKB model is not presented clearly. Some of its components, such as the components of operators and functions, have not yet had a solid structure.

In [47], a security knowledge representation artifact, called Domain Security Metamodel (DSM), is presented. This model includes a specific domain, contains information about security aspects. It is applied to the system integrating the solutions for evolution time naturally and directly. Nonetheless, this model only uses for security domains, such as the web service security domain; it does not use for knowledge domains of IPS in education.

*The algebraic approach* is a representation method based on mathematical structures. Some algebraic structures are used: groups, lattices, rings, fields, and integrating those structures [27]. By using this approach, the checking of information equivalence of knowledge is solved based on the symmetries of knowledge bases [28]. The knowledge base as logic is represented by the structure of matrices and tensors in linear algebra [29]. The structure of the knowledge base in [30] is created as a dynamic concept network mimicking human knowledge represented in the brain. This study is improved from the model of concept algebra in [31]. It is used to organize the knowledge bases of cognitive robots and machine learning systems. Nonetheless, those results are theoretical; they have not yet been applied in practical applications.

The knowledge base in [49] is represented by using the structure of relational algebra. This method is an analogical

**TABLE 1.** Comparison between methods for representing knowledge bases of an IPS in education.

| Method | Universality | Usability | Practicality | Formality |
|---|---|---|---|---|
| Formal logic | Level 2 | Level 1 | Level 1 | Level 4 |
| Networks | Level 2 | Level 3 | Level 2 | Level 3 |
| Frame | Level 2 | Level 2 | Level 2 | Level 2 |
| Ontology | Level 3 | Level 3 | Level 2 | Level 3 |
| COKB | Level 3 | Level 4 | Level 3 | Level 2 |
| Algebraic approach | Level 1 | Level 2 | Level 1 | Level 4 |

mechanism of inheritance in the Association-Oriented Database Metamodel. However, this method only can specify relationships of the knowledge base as data, and it cannot represent the real-world knowledge domain directly. Besides, there are many methods for integrating the knowledge bases in real life, such as image understanding [53], communication understanding [54]. However, those methods cannot solve problems in the learning knowledge domain. Hence, they are not sufficient to apply in IPS systems.

Methods for knowledge representation in education are compared based on criteria in Fig.1 as Table 1 [6].

## III. STRUCTURE OF RELA-OPS MODEL

Rela-Ops model is a model for representing the combining knowledge of relations and operators. It is built based on ontology and object-oriented approach, in which an object in this model has behaviors to solve problems on its own. The components of this model are sets have their properties.

*Definition 1:* The structure of Rela-Ops model includes four components:

$$\mathcal{K} = (\mathbb{C}, \mathbb{R}, \mathbb{OPS}, \mathbb{RULES})$$

In which, $\mathbb{C}$ is a set of concepts. Each concept $c \in \mathbb{C}$ is a class of objects which have their own structure and behaviors to solve problems on themselves. Each concept c also has an instance set $I_c$. $\mathbb{R}$ is a set of relations on concepts. Each relation is a binary relation between concepts in $\mathbb{C}$. This set represents the knowledge of relations in the knowledge domain $\mathcal{K}$. The $\mathbb{OPS}$-set is a set of operators. This set represents the knowledge of unary and binary operators on concepts in $\mathbb{C}$. The $\mathbb{RULES}$-set is a set of inference rules of the knowledge domain $\mathcal{K}$.

In Fig. 2: Rela-Ops model consists of four components as part (I). This model is built based on ontology. Each concept in $\mathbb{C}$ has a particular structure, and objects of each concept also have behaviors to solve problems on them, as part (II). The structure of components in the Rela-Ops model is presented in sections 3.A and 3.C. The facts of this model are defined in section 3.B. Besides, the Rela-Ops model has general problems and an inference engine to solve them as part (III). The reasoning method for solving these problems will be presented in section 4.
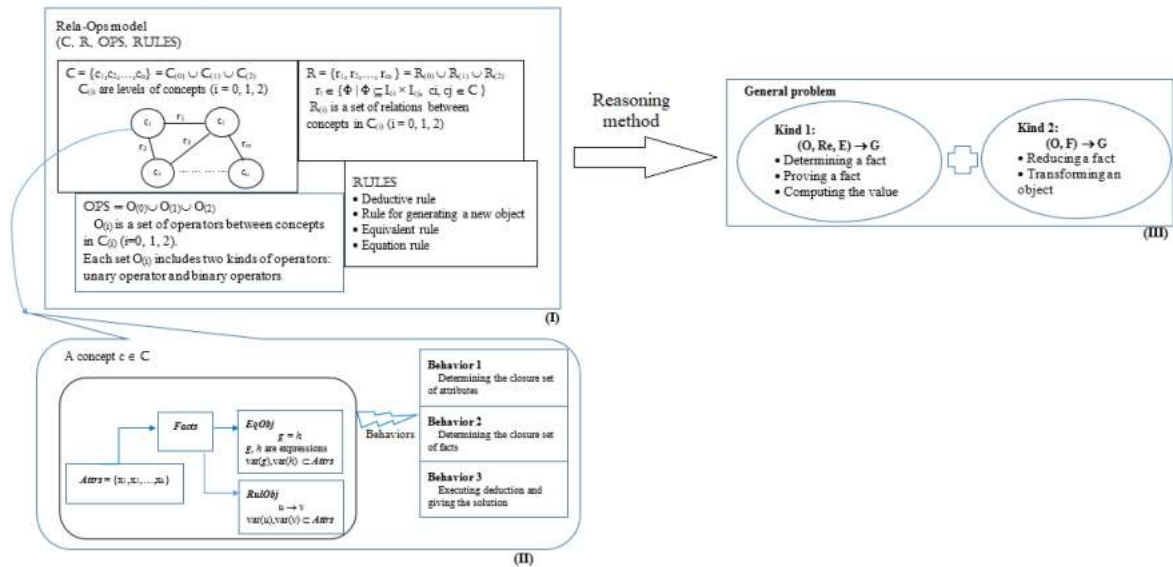
**FIGURE 2.** The structure of Rela-Ops model.

## A. STRUCTURE OF COMPONENTS ($\mathbb{C}$, $\mathbb{R}$, OPS)

The structure of each component in the Rela-Ops model is presented in Table 2.

*Example 2:* The structure of *Matrix* concept in Linear Algebra is as follows:

$Attrs = \{m, n, a[m][n], rank\}$

$m, n$: $\mathbb{N}$ // the row, column dimension of a matrix resp. *rank*: $\mathbb{N}$ // the rank of a matrix

$a[m][n]$: $\mathbb{R}$ // the elements of a matrix

$Facts := \emptyset$   $EqObj := \emptyset$

$RulObj := \{r_1: \{m = n\} \rightarrow \{this: SquareMatrix\} \}$

The structure of *SquareMatrix* concept in Linear Algebra is as follows:

SquareMatrix:: Matrix (A square matrix is a matrix)

$Attrs := $ Matrix.Attrs $\cup \{inv, diag, det, sym\}$

   *diag*: Boolean // the diagonalizable property

   *inv*: Boolean // the invertible property

   *det*: $\mathbb{R}$    // the determinant of a matrix

   *sym*: Boolean // the symmetric property

$EqObj := $ Matrix.EqObj $\cup \{m = n\}$

$RulObj := $ Matrix.RulObj $\cup$

   $\{r_1: det \# 0 \rightarrow inv = 1,$

   $r_2: \forall i, j, 1 \le i \le n, 1 \le j \le n: a[i][j] = a[j][i]$

      $\rightarrow sym = 1,$

   $r_3: \forall i, j, 1 \le i < j \le n: a[i][j] = 0$

      $\rightarrow this: UpperTriangleMatrix\}$

## B. FACTS IN RELA-OPS MODEL

### 1) SYNTAX OF A CLAUSE

*Definition 3:* Kinds of a clause in knowledge model $\mathcal{K}$.

A $\mathcal{K}$ as Rela-Ops model is a kind as follows:

| Form | Specification | Condition |
|------|---------------|-----------|
| 1 | $x:c$ | $x$ is an object, $c \in \mathbb{C}$ |
| 2 | $o$ | $o \in I_c, c \in \mathbb{C}$ |
| 3 | $o = $ \<const\> | $o \in I_c, c \in \mathbb{C}$ <br> \<const\>: constant |
| 4 | $x \Theta y$ | $\Theta \in \mathbb{R}, x \in I_{c1}, y \in I_{c2},$ <br> $c1 \in \mathbb{C}, c2 \in \mathbb{C}$ |
| 5 | \<expr1\> = \<expr2\> | \<expr1\>, \<expr2\>: expression |

Denote: $\mathcal{S} = \{p | p$ is a clause$\}$

*Definition 4:* Definition of a sentence

   a) $p \in \mathcal{S}$: $p$ is a sentence

   b) if $A$ is a sentence, so is $\neg A$

   c) if $A, B$ are sentences, so are $A \vee B, A \wedge B$.

*Definition 5:* Definition of the value of a sentence

   a) A sentence $A$ has a Boolean value, denoted Val($A$)

   b) We have a function $I$: $\mathcal{S} \rightarrow \{$true, false$\}$

   c) If $p \in \mathcal{S}$: Val(p) $= I$(p)

   d) If $A, B$ are sentences:

$$\text{Val}(A \vee B) = \text{Val}(A) \vee \text{Val}(B)$$

$$\text{Val}(A \wedge B) = \text{Val}(\wedge \text{Val}(B)$$

$$\text{Val}(\neg A) = \neg \text{Val}(A)$$

### 2) SYNTAX OF A PREDICATE

*Definition 6:* Definition of the predicate on the knowledge model $\mathcal{K}$

On a knowledge model $\mathcal{K}$, we have the predicates as followed:

1. Type$(c, x) ::= (x: c)$                ($c \in \mathbb{C}, x$ is an object)

**TABLE 2.** Structure of components.

| Level | $\mathbb{C}$ | $\mathbb{R}$ | $\mathbb{O}$PS |
|---|---|---|---|
| $\mathbb{C}_{(0)}$ | - Set of real number: $\nabla$ <br> - The basic concepts: <br>   + The basic concept c only includes an instance set. This set is denoted by $I_c$. <br>   + $I_c \neq \emptyset$ <br>   + Each $o \in I_c$ is an object of the concept c. | - Relations between real numbers $\nabla$: <br> $$\{\leq, =\}$$ <br> - $\mathbb{R}_0 \subset \{\Phi \mid \Phi \subseteq I_{c1} \times I_{c2},$ <br> $c1, c2 \in \mathbb{C}_{(0)}\}$ <br><br> * In case ci = cj, the properties of $\Phi$ are considered: reflexive, symmetric, asymmetric, and transitive. | - Operators between real numbers in $\nabla$. <br> - Operators between concepts in $\mathbb{C}_{(0)}$: <br> $$O_{(0)} = O^1_{(0)} \cup O^2_{(0)}$$ <br> In which: <br> + Set of unary operators $O^1_{(0)}$: <br> $O^1_{(0)} \subset \{\oplus: I_{c1} \to I_{c2} \mid$ <br> $c1, c2 \in \mathbb{C}_{(0)}\}$ <br> + Set of binary operators $O^2_{(0)}$: <br> $O^2_{(0)} = \{\otimes: I_{c1} \times I_{c2} \to I_{c3} \mid$ <br> $c1, c2, c3 \in \mathbb{C}_{(0)}\}$ |
| $\mathbb{C}_{(1)}$ | **(Attrs, Facts, EqObj, RulObj)** <br> 1/ *Attrs* is a set of attributes: <br> $\emptyset \neq$ Attrs $\subset \{ x_i, i = 1..n \mid$ <br> $x_i \in I_{ci}, ci \in \mathbb{C}_{(0)}\}$ <br> 2/ *Facts* is a set of facts between attributes: <br> Facts $\subset \{f \mid f$ is a fact, <br> var($f$) $\subseteq$ Attrs$\}$ <br><br> 3/ *EqObj* is a set of equations between of attributes: <br> EqObj $\subset \{u = v \mid$ <br> $u, v$ are expressions, <br> var($u$) $\subseteq$ Attrs, var($v$) $\subseteq$ Attrs$\}$ <br><br> 4/ *RulObj* is a set of deductive rules: <br> RulObj $\subset \{ p \to q \mid p \subseteq$ Attrs, <br> $q \subseteq$ Attrs, $p * q = \emptyset\}$ | $\mathbb{R}_1 \subset \{\Phi \mid \Phi \subseteq I_{c1} \times I_{c2},$ <br> $c1, c2 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)}$ <br> $c1 \in \mathbb{C}_{(1)} \vee c2 \in$ <br> $\mathbb{C}_{(1)}$ <br> $\}$ <br> * In case ci = cj, the properties of $\Phi$ are considered: reflexive, symmetric, asymmetric and transitive. | - Operators between concepts in $\mathbb{C}_{(0)}$ and $\mathbb{C}_{(1)}$: <br> $$O_{(1)} = O^1_{(1)} \cup O^2_{(1)}$$ <br> In which: <br> + Set of unary operators $O^1_{(1)}$: <br> $O^1_{(1)} \subset \{\oplus: I_{c1} \to I_{c2} \mid$ <br> $c1, c2 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)}\}$ <br> + Set of binary operators $O^2_{(1)}$: <br> $O^2_{(1)} \subset \{ \otimes: I_{c1} \times I_{c2} \to I_{c3} \mid$ <br> $c1, c2, c3 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)}\}$ <br> Some properties of an operator are checked: commutation, association, identity. |
| $\mathbb{C}_{(2)}$ | **(Attrs, Facts, EqObj, RulObj)** <br> 1/ $\emptyset \neq$ Attrs $\subset \{x_i, i = 1..n \mid x_i \in I_{ci},$ <br> $ci \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)}\}$ <br> 2/ $\exists\, a \in$ Attrs, $\exists c_a \in \mathbb{C}_{(1)}, a \in I_{ca}$ <br><br> 3/ Facts $\subset \{f \mid f$ is a fact, <br> var($f$) $\subseteq$ Attrs$\}$ <br> 4/ EqObj $\subset \{u = v \mid$ <br> $u, v$ are expressions, <br> var($u$) $\subseteq$ Attrs, var($v$) $\subseteq$ Attrs$\}$ <br><br> 5/ RulObj $\subset \{ p \to q \mid p \subseteq$ Attrs, <br> $q \subseteq$ Attrs, $p * q = \emptyset\}$ | $\mathbb{R}_2 \subset \{\Phi \mid \Phi \subseteq I_{c1} \times I_{c2},$ <br> $c1, c2 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)} \cup$ <br> $\mathbb{C}_{(2)}$ <br> $c1 \in \mathbb{C}_{(2)} \vee c2 \in \mathbb{C}_{(2)} \}$ <br><br> * In case ci = cj, the properties of $\Phi$ are considered: reflexive, symmetric, asymmetric, and transitive. | - Operators between concepts in $\mathbb{C}_{(0)}, \mathbb{C}_{(1)},$ and $\mathbb{C}_{(2)}$: <br> $$O_{(2)} = O^1_{(2)} \cup O^2_{(2)}$$ <br> In which: <br> + Set of unary operators $O^1_{(2)}$: <br> $O^1_{(2)} \subset \{ \oplus: I_{c1} \to I_{c2} \mid$ <br> $c1, c2 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)} \cup \mathbb{C}_{(2)}\}$ <br> + Set of binary operators $O^2_{(2)}$: <br> $O^2_{(2)} \subset \{ \otimes: I_{c1} \times I_{c2} \to I_{c3} \mid$ <br> $c1, c2, c3 \in \mathbb{C}_{(0)} \cup \mathbb{C}_{(1)} \cup \mathbb{C}_{(2)}\}$ <br> Some properties of an operator are checked: commutation, association, identity. |

2. Determine($x$)::= $o$      ($c \in \mathbb{C}, x \in I_c$)

3. Equal$_{const}$($x$)::= ($x = $ \<const\>)      ($c \in \mathbb{C}, x \in I_c$, \<const\>: constant)

4. Rela$_\Theta$($x, y$)::= ($x \Theta y$)      ($\Theta \in \mathbb{R}$, c1, c2 $\in \mathbb{C}$, $x \in I_{c1}, y \in I_{c2}$)

5. Equal($e_1, e_2$)::= ($e_1 = e_2$)      ($e_1, e_2$ are expressions)

Let $\mathcal{P}_K = \{f \mid f$ is a predicate$\}$

  $\alpha_K: \mathcal{P}_K \to \mathbb{N}$: assigning arities to a predicate.

*Definition 7:* Definition of a sentence

  a) $p \in \mathcal{S}$: $p$ is a sentence

b) if $f \in \mathcal{P}_K, \alpha_K(f) = n$ and $x_1, x_2, \ldots, x_n$ are variables, then $f(x_1, x_2, \ldots, x_n)$ is a sentence.

c) if A is a sentence, so is $\neg A$

d) if A, B are sentences, so are $A \vee B, A \wedge B$.

e) if A is a sentence and $x$ is an individual variable, so are $(\forall x)$A, $(\exists x)$A.

*Definition 8:* Definition of the value of a sentence

  a) if $p \in \mathcal{S}$: Val($p$) = $I(p)$

  b) if $f \in \mathcal{P}_K, \alpha_K(f) = $ n and $x_1, x_2, \ldots, x_n$ are variables:

$$Val(f(x_1, x_2, \ldots, x_n)) = I(f(x_1, x_2, \ldots, x_n))$$

c) if *A* is a sentence and *x* is an invidual variable:

Val($\forall xA$)

$$= \begin{cases} \text{true} & \text{if Val }(\lambda_{x=d}(A)) = \text{true for all variables } d \\ & \quad (d \text{ has the type A's arguments)} \\ \text{false} & \text{otherwise} \end{cases}$$

Val($\exists xA$)

$$= \begin{cases} \text{true} & \text{if Val }(\lambda_{x=d}(A)) = \text{true for some variables } d \\ & \quad (d \text{ has the type A's arguments)} \\ \text{false} & \text{otherwise} \end{cases}$$

$\lambda_{x=d}(A)$ : is substituted *d* for *x* in *A*

d) if *A, B* are sentences:

$$\text{Val}(A \vee B) = \text{Val}(A) \vee \text{Val}(B)$$
$$\text{Val}(A \wedge B) = \text{Val}(A) \wedge \text{Val}(B)$$
$$\text{Val}(\neg A) = \neg \text{Val}(A)$$

### 3) UNIFICATION OF FACTS

*Definition 9:*

a) A *fact* in Rela-Ops model is a sentence as clause form (definition 3.3) or predicate form (definition 3.6).

b) Let *f* and *g* be two facts. The unification of *f* and *g*, denoted $f \cong g$, is defined as the following conditions:

1. *f* and *g* are clauses and have the same form *k*, with $1 \le k \le 5$: use the unification of facts in [9], [10].
2. OR if $f = \neg f_1$ and $g = \neg g_1$, and $f_1, g_1$ are facts:

$$f_1 \cong g_1.$$

3. OR if $f = f_1 \wedge f_2$ and $g = g_1 \wedge g_2$, and $f_1, f_2, g_1, g_2$ are facts:

$$(f_1 \cong g_1 \text{ and } f_2 \cong g_2) \text{ or } (f_2 \cong g_1 \text{ and } f_2 \cong g_1).$$

4. OR if $f = f_1 \vee f_2$ and $g = g_1 \vee g_2$, and $f_1, f_2, g_1, g_2$ are facts:

$$(f_1 \cong g_1 \text{ and } f_2 \cong g_2) \text{ or } (f_2 \cong g_1 \text{ and } f_2 \cong g_1).$$

5. OR if $f = \forall o_1 f_1$ and $g = \forall o_2 g_1$, and $f_1, g_1$ are facts, $o_1, o_2$ are objects:

$$(o_1 \cong o_2 \text{ and } f_1 \cong g_1).$$

6. OR if $f = \exists o_1 f_1$ and $g = \exists o_2 g_1$, and $f_1, g_1$ are facts, $o_1, o_2$ are objects:

$$(o_1 \cong o_2 \text{ and } f_1 \cong g_1).$$

*Definition 10 [10]: Relations and operators on a set of facts:*

Let A and B be sets of facts, and *f* be a fact, the definition of relations and operators between them as follows:

$$f \odot A \Leftrightarrow \exists g \in A, f \cong g \qquad A \sqcap B = \{f | f \odot A - f \odot B\}$$
$$A \sqsubseteq B \Leftrightarrow \forall f \in A, f \odot B \qquad A \sqcup B = \{f | f \odot A.f \odot B\}$$
$$A \cong B \Leftrightarrow A \sqsubseteq B \wedge B \sqsubseteq A \quad A \backslash B = \{f | f \odot A - not(f \odot B)\}$$

**TABLE 3.** The kinds of rules in Rules-set.

| **Rule$_{deduce}$** | **Rule$_{generate}$** | **Rule$_{equivalent}$** | **Rule$_{equation}$** |
|---|---|---|---|
| r is a *deductive rule*, and it has the form: $h(r) \longrightarrow g(r)$ where *h(r)*, *g(r)* are sets of facts. | r is a deductive rule *for generating a new object*, and it has the form as follows: $h(r) \longrightarrow g(r)$ where *h(r)* and *g(r)* are sets of facts, they satisfy conditions: $\exists$ object *x*, $x \in g(r)$ and $x \notin h(r)$ | r is an *equivalent rule*, it has form: $f(r), h(r) \leftrightarrow g(r)$ where *f(r), h(r)* and *g(r)* are sets of facts, they satisfy conditions: $f(r), h(r) \longrightarrow g(r)$, and $f(r), g(r) \longrightarrow h(r)$ are true. | r is an *equation rule*, and it has the form: $u = v$ where *u, v* are expressions of objects. Denote: left(r) = *u* right(r) = *v* |

### C. STRUCTURE OF RULES-SET

A rule r $\in$ RULES is one of the four kinds below:

$$\text{RULES} = \textbf{Rule}_{\textbf{deduce}} \cup \textbf{Rule}_{\textbf{generate}}$$
$$\cup \textbf{Rule}_{\textbf{equivalent}} \cup \textbf{Rule}_{\textbf{eqnarray}}$$

In the Rela-Ops model, the structure of each component has been built completely. The syntax of a clause and a predicate is defined clearly, thought that the facts and their unification are studied. Based on the structure of this model, the model of problems and algorithms for solving them will be presented in the next section.

### D. SPECIFICATION LANGUAGE

**Definition of concept:**

```
concept-def::= CONCEPT <name>
                    [isa]
                    [attributes]
                    [facts]
                    [equations]
                    [rules]
ENDCONCEPT;
isa::=<name>:type
```

**Definitions of relations:**

```
relation-def::= RELATION <name>
            ARGUMENT: argument-def +;
            [facts]
            [prob-type]
ENDRELATION;
argument-def::= name
prob-type::= reflexive | symmetric | asymmetric | transitive.
```

**Definitions of operators:**

```
Operator-def: = OPERATOR <name>
            ARGUMENT: argument-def+
            RETURN: return-def+;
            PROPERTY: prob-type+
            [constraint]
            [variables]
             [statements]
        END OPERATOR.
argument-def: =<name>: type
return-def: = name: type
prob-type: = commutative|associative|identity
```

statement: = name: =<expression>.
**Definitions of rules:**
    rule-def::= RULE <name>
            [kind]
            OBJECT: argument-def $^+$;
            [facts]
            HYPOTHESIS:
                    [facts]
            GOAL:
                    [facts]
    ENDRULE;
    kind::= deductive | generate | equivalent |equation.

In practice, a knowledge domain $\mathcal{K}$ as the Rela-Ops model can be represented by a restriction model. These restriction models can reduce the complexity of an ordinary model. The knowledge of relations is a popular kind of knowledge. In this kind of knowledge domains, the computation is only the simple computing on real values, so the model for this knowledge does not need the component representing operators. A model of knowledge of relations is $(\mathbb{C}, \mathbb{R}, \mathbb{RULES})$ as a restriction model as Rela-Ops model lacking $\mathbb{OPS}$-set, this kind of model was studied in [9], [11]. Besides, the knowledge of operators is also used in the computation knowledge domains. In these kinds of knowledge domains, the relations between objects are often as computational relationships. The general problems in those knowledge domains are solving the equations, transforming the expression between objects. A model of knowledge of relations is $(\mathbb{C}, \mathbb{OPS}, \mathbb{RULES})$ as a restriction model as Rela-Ops model lacking R-set, this kind of model was studied in [10], [12].

## IV. MODEL OF PROBLEMS AND ALGORITHMS ON RELA-OPS MODEL

There are two kinds of problems on the Rela-Ops model: Problems on an object and general problems on the model. Problems on an object are its behaviors, and they are solved based on the reasoning on its structure. Model and the solving method of this kind of problem are presented in [9], [10]. In this section, we only mention the model and solving methods for general problems.

### A. MODELS OF GENERAL PROBLEMS

*Definition 11:* There are two kinds of general problems. Models of them are as follows:
    a/ <u>Kind 1</u>: Problems can be represented by the form:

$$(\mathbf{O}, \mathbf{Re}, \mathbf{E}) \rightarrow \mathbf{G}$$

where,
    $O = \{O_1, O_2 \ldots, O_m\}$: set of objects in the problem.
    $Re = \{r_1, r_2, \ldots, r_n\}$: set of relations between objects in O.
    $E = \{e_1, e_2, \ldots, e_k\}$: set of equations.
    $G = \{$"KEYWORD": $g\}$ with "KEYWORD" is a keyword of the goal and $g$ is a sentence, "KEYWORD" may be the followings:

- "Determine": this goal is determining the sentence $g$.
- "Prove": this goal is proving a sentence $g$.
- "Compute": this goal is determining the value of the expression $g$.
    b/ <u>Kind 2</u>: Problems can be represented by the form:

$$(\mathbf{O}, \mathbf{F}) \rightarrow \mathbf{G}$$

where, $F = \{f_1, f_2, \ldots, f_p\}$: set of facts.
    $G = \{$"KEYWORD": $g\}$ with "KEYWORD" may be the followings:
- "Reduce": this goal is reducing the expression $g$.
- "Transform": this goal is transforming an object $g$ into an expression between certain objects.

*Definition 12:* Let $\mathcal{K}$ be a knowledge domain as Rela-Ops model, and a problem $P = (\mathbf{O}, \mathbf{Re}, \mathbf{E}) \rightarrow \mathbf{G}$ as kind 1. Suppose $S = [s_1, s_2, \ldots, s_k]$ is a list of rules.

Denote: $E_0 = E$, $E_1 = s_1(E_0)$, $E_2 = s_2(E_1)$, ...,

$$E_k = s_k(E_{k-1}) \text{ and } S(E) = E_k.$$

where $s_i(E_{i-1})$ is the set facts can be deduced from $E_{i-1}$ by rule $s_i$ $(1 \le i \le k)$.

A problem P is *solvable* if and only if there exists a list S such that $\mathbf{G}.g \odot S(E)$.

### B. ALGORITHMS FOR SOLVING PROBLEMS
#### 1) ALGORITHM FOR SOLVING A PROBLEM IN-KIND 1

*Algorithm 1:* Let $\mathcal{K} = (\mathbb{C}, \mathbb{R}, \mathbb{OPS}, \mathbb{RULES})$ be a knowledge domain as Rela-Ops model, and a problem $P = (\mathbf{O}, \mathbf{Re}, \mathbf{E}) \rightarrow \mathbf{G}$ as kind 1 in Def. 11. This algorithm will solve the problem P though these steps as follows:
    **Input:** The problem $P = (\mathbf{O}, \mathbf{Re}, \mathbf{E}) \rightarrow \mathbf{G}$
    **Output:** The solution to problem P.

The method of the following general algorithm uses forward chaining reasoning. It also uses heuristics rules in the reasoning process of searching for applied rules. Objects attend this process as active agents. We use the characteristic of the relations and operators to get new facts. This processing is done when it gets the goal.

#### 2) ALGORITHM FOR SOLVING A PROBLEM IN-KIND 2

*Algorithm 2:* Let $\mathcal{K} = (\mathbb{C}, \mathbb{R}, \mathbb{OPS}, \mathbb{RULES})$ be a knowledge domain as Rela-Ops model, and $P = (\mathbf{O}, \mathbf{F}) \rightarrow \mathbf{G}$ be a problem as kind 2 in Def. 11.
    **Input:** The problem $P = (\mathbf{O}, \mathbf{F}) \rightarrow \mathbf{G}$
    **Output:** The solution to problem P.

The algorithm for solving this kind of problem uses a forward chaining strategy to get new facts. The reasoning combines heuristic rules to make the deducing more effectively. This algorithm was studied and designed in [9], [10], [48]. It only needs to change about combining the facts of relations and equations in reasoning processing.

**Algorithm 1**

**Step 0:** Initialize variables

*flag*:= true;

*KnownFacts*:= **Re** ⊔ **E**;

*count*:=0; # the number of new objects which are generated

*Sol*:=[ ]; # solution of problem

**Step 1.** Collect objects in hypothesis and goal part. Classify kind of facts in **Re** and **E**.

**Step 2.** Check **G**.

If **G** is achieved **then**

**Go to** step 5.

**Step 3:** Determine the closure of each object in **O** by using its behaviors and facts in **Re** and **E**.

**Step 4:** Use equations in **E** to generate the new facts as relation form.

Use the relations in **Re** to generate new equations.

Update *KnownFacts*.

**Step 5:** Select a rule in set **Rules** to produce new facts or new objects by using heuristic rules.

**while** (*flag* != false) and **not**(**G** is determined) **do**

Search *r* in **Rules** which can be applied to KnownFacts

**5.1.** Case: *r is a deductive rule*

**if** (*r* has form: $h(r) \rightarrow g(r)$) **then**

*KnownFacts*:= *KnownFacts* +$g(r)$;

*s*:=[$r, h(r), g(r)$];

*Sol*:=[op(*Sol*), *s*];

continue;

**end if;**

**5.2.** Case: *r is a rule for generating a new object*

# *r* has form: $h(r) \rightarrow g(r)$

**if** *count* ≤ card(**O**) **then** #only generate at most number of objects in hypothesis

**if** (*r* generates a new object *o*) and

not($o \odot KnownFacts$) **then**

*count*:=*count*+1;

*KnownFacts*:= *KnownFacts* +$g(r)$;

*s*:=[$r, h(r), g(r)$];

*Sol*:=[op(*Sol*), *s*];

**Goto** Step 3 with new object *o*;

**end if;**

**end if;** #5.2

**5.3.** Case: *r is an equivalent rule*

**if** (*r* has form: $f(r), h(r) \leftrightarrow g(r)$ ) **then**

*KnownFacts*:= *KnownFacts* +$g(r)$;

*s*:=[$r, h(r), g(r)$];

*Sol*:=[op(*Sol*), *s*];

continue;

**end if;** #5.3

**5.4.** Case: *r is an equation rule*

**if** (*r* has form: $u = v$) **then**

*r* can generate set of new facts A =

$r(KnownFacts) = \{f | f$ is a fact of kind 2 or kind 5\}

*s*:=[*r, KnownFacts*, A];

*Sol*:=[op(*Sol*), *s*];

**Algorithm 1** *(Continued.)*

**if** (*r* generates a new object *o*) and

not($o \odot KnownFacts$) **then**

*count*:=*count*+1;

*KnownFacts*:= *KnownFacts* +A;

**Goto** Step 3 with new object *o*;

**else**

*KnownFacts*:= *KnownFacts* +A;

continue;

**end if;**

**end if;** #5.4

**5.5. if** (rule *r* cannot be found) **then**

*flag*:= false;

**end if;**

**end do**; #while

**Step 6:** Conclusion of problem

**if G** is determined **then**

Problem **(O, Re, E)**→**G** is solvable;

*Sol* is a solution of problem;

Reduce *Sol* by eliminating redundant rules.

**else**

Problem **(O,Re,E)**→**G** is unsolvable;

**end if;**

## C. THEOREMS

*Theorem 13:* Algorithm 1 is finite; it stops after finite steps.

*Proof:*

+ As this algorithm, the number of new objects that can be generated in Step 5.2 does not exceed the card(**O**), and the number of objects in set **O** is finite.

+ The number of deductive rules in RULES-set is finite; thus, the number of new facts which are deduced in Step 5.1 is finite.

+ The number of equivalent rules in RULES-set is finite, and each expression has finite transforming steps in Step 5.3. Besides, the problem only has finite expressions; thus, step 5.3 will be stopped after finite steps.

+ The number of equations rules in RULES-set is finite; thus, the number of new objects that can be generated in Step 5.4 is finite.

So this algorithm gives the conclusion after finite steps.

*Lemma 14:* Let $\mathcal{K}$ be a knowledge domain as Rela-Ops model and **(O, Re, E)** be the hypothesis of a problem as kind 1. By theorem 4.1, there exists a unique set $\mathcal{L}_{(O,Re,E)}$ such that it is the maximum set containing all facts which can be deduced from **(O, Re, E)**.

*Theorem 15:* Let $\mathcal{K}$ be a knowledge domain as Rela-Ops model and a problem P = **(O, Re, E)** → **G** as kind 1. These statements are equivalent:

(i) Problem P is solvable.

(ii) $\mathbf{G}.g \odot \mathcal{L}_{(O,Re,E)}$

**Algorithm 2**

**Step 0:** Initialize variables

*Known*:= F //Set of facts can be reasoned

*Sol*:=[ ] // the solution of this problem

**Step 1:** Classify (O, F) by the knowledge domains

$O_p$, $F_p$: Set of objects and set of facts only belong to the knowledge domain $\mathcal{K}_p$.

$O_q$, $F_q$: Set of objects and set of facts only belong to the knowledge domain $\mathcal{K}_q$.

$F_{pq}$: The set of facts between the objects belongs to both $\mathcal{K}_p$ and $\mathcal{K}_q$.

**Step 2:**

**Use** algorithm 2 to deduce the new facts based on $(O_p, F_p)$ in the knowledge domain $\mathcal{K}_p$.

**Update** *Known* và *Sol*;

**Step 3:**

**Use** algorithm 2 to deduce the new facts based on $(O_q, F_q)$ in the knowledge domain $\mathcal{K}_q$.

**Update** *Known* và *Sol*;

**Step 4:**

Use the rules $\mathbb{CONNECT}$ the knowledge of $\mathcal{K}_p$ and $\mathcal{K}_q$ in Connect for generating the new objects only belonging to a knowledge domain ($\mathcal{K}_p$ or $\mathcal{K}_q$).

$O_{pq}$: a set of new objects.

**Step 5:** Suppose that the objects in $O_{pq}$ belong to $\mathcal{K}_q$.

**Use** the objects in $O_{pq}$ and the facts in $F_{pq}$ and *Known* for **transforming** the problem P to the problem P' on $\mathcal{K}_q$.

**Step 6:**

**Solve** the problem P' on the knowledge domain $\mathcal{K}_q$ by using algorithm 1.

**Update** *Known* and *Sol*.

**Step 7:**

**If** problem P' is solvable

Problem P has a solution *Sol*.

**Else** There is no solution for problem P.

---

(iii) There exists a list of rules S = $[s_1, s_2, \ldots, s_k]$ such that **G**.$g \odot$ S(E)

*Proof:*

- (i) and (iii) are equivalent by Def. 12.
- (i) $\Rightarrow$ (ii): Problem (**O, Re, E**) $\rightarrow$ **G** is solvable, but by lemma 14, $\mathcal{L}_{(O,Re,E)}$ is a maximum set containing all facts that can be deduced from (**O, Re, E**), so **G**.$g \odot \mathcal{L}_{(O,Re,E)}$.
- (ii) $\Rightarrow$ (iii): $\mathcal{L}_{(O,Re,E)}$ is a set of all facts which can be deduced from (**O, Re, E**), then by Def. 3, there exists a list of rules S = $[s_1, s_2, \ldots, s_k]$ such that S(E) = $\mathcal{L}_{(O,Re,E)}$. But **G**.$g \odot \mathcal{L}_{(O,Re,E)}$ so **G**.$g \odot$ S(E)

Algorithm 1 is designed by using forward chaining, and theorem 15 shows that goals of problems would be deduced by a forward chaining reasoning. Therefore, the theorem 15 ensures the effectiveness of algorithm 1.

## V. KNOWLEDGE MODEL FOR MULTIPLE KNOWLEDGE DOMAINS

### A. KNOWLEDGE MODEL FOR MULTIPLE KNOWLEDGE DOMAINS

In practice, a knowledge domain $\mathcal{K}$ can include multiple knowledge sub-domains $\mathcal{K}_i$ ($i = 1, 2, \ldots, n$). Each sub-domain has a known form, and they have certain relationships between them. In this paper, each sub-domain $\mathcal{K}_i$ is modeled by using the Rela-Ops model or its reduced model, so we can establish a knowledge model M($\mathcal{K}_i$) of knowledge $\mathcal{K}_i$. There are also relationships on $\{\mathcal{K}_i\}$; thus, models $\{M(\mathcal{K}_i)\}$ and their relations are also specified. We determine a knowledge model M($\mathcal{K}$) for the whole knowledge, $\mathcal{K}$.

*Definition 16:* The model for multiple knowledge domains is a tube:

$$(\mathbb{K}, \mathbb{CONNECT})$$

In which, $\mathbb{K} = \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n\}$ is a set of knowledge sub-domains, and each sub-domain has the form as the Rela-Ops model. Connect is a set of connective rules between sub-domains $\mathcal{K}_i$ ($i = 1, 2, \ldots, n$). In this paper, we only consider the rules for transforming the knowledge in $\mathcal{K}_i$ into the knowledge in $\mathcal{K}_j$ ($j \neq i; i, j = 1 \ldots n$).

### B. PROBLEMS ON THE MODEL FOR MULTIPLE KNOWLEDGE DOMAINS

Let a knowledge domain $\mathcal{K} = (\mathbb{K}, \mathbb{CONNECT})$, including knowledge sub-domains $\mathcal{K}_i \in \mathbb{K}$ ($i = 1 \ldots n$). The general problem on the Rela-Ops model can solve problems that only use the knowledge of a knowledge domain $\mathcal{K}_i$. However, it cannot solve the problems which require the combination of multiple knowledge domains for solving. In this section, problems about combining the knowledge domains have been studied and solved. The model of these problems is also as: (**O, F**)$\rightarrow$**G**. It is classified into two kinds:

● Kind 1: The hypothesis of the problem has objects which belong to knowledge domains $\mathcal{K}_p$ and $\mathcal{K}_q$ ($p \neq q$; $p, q = 1 \ldots n$)

$$\forall o \in O, \quad \exists c_o \in \mathcal{K}_p \cdot \mathbb{C} \cup \mathcal{K}_q \cdot \mathbb{C} : o \in I_{c_o}$$

● Kind 2: The hypothesis of the problem only has objects which belong to one knowledge domain $\mathcal{K}_p$ ($p = 1 \ldots n$)

$$\forall o \in O, \quad \exists c_o \in \mathcal{K}_p \cdot \mathbb{C} : o \in I_{c_o}$$

Both kinds of problems require using the knowledge in the other knowledge domains to solve them.

*Algorithm 3: Solving a problem in kind 1*

Given a knowledge domain $\mathcal{K} = (\mathbb{K}, \mathbb{CONNECT})$ including knowledge sub-domains $\mathcal{K}_i \in \mathbb{K}$ ($i = 1 \ldots n$), and P = (O, F) $\rightarrow$ G is a problem, P has objects which belong to knowledge domains $\mathcal{K}_p$ and $\mathcal{K}_q$ ($p \neq q; p, q = 1 \ldots n$)

**Input:** The problem P = (O, F) $\rightarrow$ G such that:

$$\forall o \in O, \exists c_o \in \mathcal{K}_p \cdot \mathbb{C} \cup \mathcal{K}_q \cdot \mathbb{C} : o \in I_{c_o}$$

**Output:** The solution to problem P.

**Algorithm 3**

**Step 0:** Initialize variables

    *Known*:= F //Set of facts can be deduced.

    *Sol*:=[ ] // the solution of this problem

    $n$:= | K | // number of the sub-domains in K

**Step 1:** Generate new facts based on (O, F) by using algorithm 2 on the knowledge domain $\mathcal{K}_p$.

    **Update** *Known* and *Sol*.

**Step 2:**

    Domain:= [$p$];

    prev:= $p$;

    **do{**

      **2.1 Determine** the knowledge domain $\mathcal{K}_q$ related to $\mathcal{K}_{\text{prev}}$ though the connective rules in $\mathbb{CONNECT}$ ($q \notin$ Domain)

    Domain:= [op(Domain), $q$];

    **Generate** the new objects in $\mathcal{K}_q$ that related to $\mathcal{K}_{\text{prev}}$.

      O$_q$: a set of new objects.

      **2.2 Use** the objects in O$_q$ and the facts in *Known* for **transforming** the problem P to the problem P' on $\mathcal{K}_q$.

      **Update** *Known* và *Sol*;

      **2.3 Solve** the problem P' on the knowledge domain $\mathcal{K}_q$ by using algorithm 1.

      **Update** *Known* and *Sol*.

      prev:= q;

    **}While** (Problem P' is unsolvable) **and** (|Domain| $\leq n$)

**Step 3:**

  **if** P' is solvable

    **for** $i$ from |Domain|-1 **down to** 1 **do**

      prev:=Domain[$i$];

      **Transform** results into the knowledge domain $\mathcal{K}_{\text{prev}}$.

        **Update** *Known* and *Sol*.

    **end do;** #for

    Problem P has a solution *Sol*.

  **Else**

    There is no solution to problem P.

*Algorithm 4: Solving a problem in kind 2*

Given a knowledge domain $\mathcal{K} = (\mathbb{K}, \mathbb{CONNECT})$ including knowledge sub-domains $\mathcal{K}_i \in \mathbb{K}$ ($i = 1 \ldots n$), and P = (O, F) $\rightarrow$ G is a problem, P only has objects which belong to one knowledge domain $\mathcal{K}_p$ ($p = 1 \ldots n$).

**Input:** The problem P = (O, F) $\rightarrow$ G such that:

$$\forall o \in O, \exists c_o \in \mathcal{K}_p \cdot \mathbb{C} : o \in I_{c_o}$$

**Output:** The solution to problem P.

# VI. APPLICATION FOR DESIGNING INTELLIGENT PROBLEMS SOLVER SYSTEMS IN MATHEMATICS

## A. DESIGN AN INTELLIGENT SYSTEM FOR SOLVING PROBLEMS IN LINEAR ALGEBRA

Linear Algebra is a required course in the university. In this course, chapters about matrix, linear system, and vector space are the foundation of computational techniques. They help students improve their basic knowledge about mathematics, solving the problems. The IPS for Linear Algebra has to solve common exercises in this course, including basic and advanced kinds. Moreover, solutions to this system have to satisfy the requirements of an IPS. They have to ensure the criterion about pedagogy. They are readable, step-by-step, especially their reasoning simulates the method for solving of students.

There are many current programs for solving exercises in Linear Algebra, but they have not yet tended to learn supporting systems, some requirements in education are missing. Symbolab [39] and Wolfram|Alpha [41] are websites for solving problems step-by-step; however, their knowledge bases are organized as frames, so they cannot be used to solve the problems that require in-depth knowledge. Maple [42] and Matlab [43] are computer algebra systems; their computation on linear algebra is fast; however, they do not have a reason to get solutions to problems similar to the solving method of learners.

In this section, we build an IPS system in Linear Algebra at university. In this course, chapters about Matrices, Linear equations system, and Vector space are essential for students. Our system can solve some kind of exercise in these chapters. Its knowledge base is represented by the knowledge model for multiple knowledge domains:

$$(\mathbb{K}, \mathbb{CONNECT})$$

### 1) $\mathbb{K}$ – SET OF SUB-DOMAINS

In this system, the knowledge domain about Linear Algebra is collected from [32], [33]. It is partitioned into three knowledge sub-domains:

$$\mathbb{K} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$$

In which:

• $\mathcal{K}_1 = (\mathbb{C}_1, \mathbb{R}_1, \mathbb{OPS}_1, \mathbb{RULES}_1)$ is a knowledge domain about Matrices - Vectors.

• $\mathcal{K}_2 = (\mathbb{C}_2, \mathbb{R}_2, \mathbb{RULES}_2)$ is a knowledge domain about Linear equations systems.

• $\mathcal{K}_3 = (\mathbb{C}_3, \mathbb{R}_3, \mathbb{OPS}_3, \mathbb{RULES}_3)$ is a knowledge domain about Vector spaces.

The detail of these knowledge bases is represented in Appendix A.

### 2) $\mathbb{CONNECT}$ – SET OF CONNECTIVE RULES BETWEEN THE SUB-DOMAINS

This set includes the rules for transforming the knowledge in a sub-domain $\mathcal{K}_i \in \mathbb{K}$ into the other sub-domains $\mathcal{K}_j$ ($j \neq i$; $i, j = 1, 2, 3$). The connection of the knowledge $\mathcal{K}_1$ and $\mathcal{K}_2$ is the equivalent transforming between the set of roots of two linear equations systems and the augmented matrices of two corresponding linear equations systems. This transforming converts the problems in the knowledge $\mathcal{K}_2$ into the problems in the knowledge $\mathcal{K}_1$ by rule 2.2 in Appendix A.
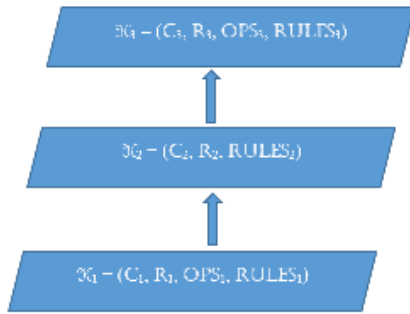
**FIGURE 3.** The relations between the knowledge sub-domains in Linear Algebra.
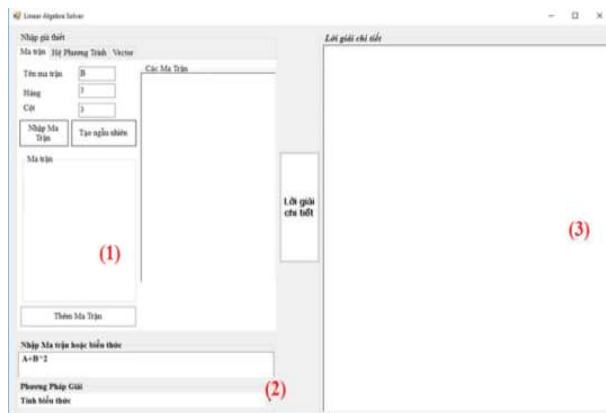


**FIGURE 4.** The user interface of our system. (1) Input objects and facts of the exercise: They are the hypothesis (O, F) of the exercise. They are input by the specification language of the program. (2) Requirements of the exercise (goal): Input the goal of the exercise. (3) Solution: Show the solution of the exercise.

Based on the result "*The set of roots of a linear equations system with n variables is a vector subspace of* $\mathbb{R}$", we have the connection between the knowledge $\mathcal{K}_3$ and $\mathcal{K}_2$ through the finding a spanning-set and proving the linear independence of a set of vectors in the knowledge $\mathcal{K}_3$ by solving a linear equations system in the knowledge $\mathcal{K}_2$. The transforming converts the problems in the knowledge $\mathcal{K}_3$ into the problems in the knowledge $\mathcal{K}_2$ by rules 3.2 and 3.3 in Appendix A.

Fig. 3 represents the relations between the those knowledge sub-domains.

### 3) TESTING AND EXPERIMENTS

Our system can solve the typical kinds of basic and advanced problems for this course. Its solutions are clear, readable, and their reasoning is similarly the method for solving of students.

Firstly, the exercise is inputted to our program using the model of problems **(O, F)**→**G** as section 5.2. Our computer program has the specification language to represent the exercise. The inference engine of our system is designed based on algorithms for solving problems on the knowledge model for multiple knowledge domains as section 5.2. Then, the program solves this exercise automatically. Its solution trends to meet the requirements of an IPS in education [48]. Fig. 4 is the user-interface of our system:

**TABLE 4.** Kinds of problems can be solved.

| Chapter | Symbolab | Our system |
|---|---|---|
| Matrix-Vector | Solve basic and advanced problems, such as:<br>+ Transforming and computing an expression between matrices.<br>+ Determining the determinant, rank, inverse of a matrix.<br>+ Computing the eigenvalues and eigenvectors of a matrix.<br>+ Diagonalizing a matrix. | |
| | Only solve problems about computing the determinant, rank with non-parameter. | Can solve problems about computing the determinant, rank with parameter. |
| Linear Equations system | + Solving a linear equations system.<br>+ Solving a simple matrix equation that can be transformed into a linear equation system. | |
| | Only solve a linear equation system with non-parameter. | Can solve a linear equations system with parameter. |
| Vector space | It cannot solve problems with vector spaces. | Solve some basic kinds of problems:<br>+ Find the based-set of a vector space.<br>+ Determine a vector space from a spanning-set of vectors.<br>+ Compute the direct sum of subspaces. |

Symbolab is a website that can automatically solve mathematical problems. It gives step-by-step solutions [37]. It can solve some problems with matrices and equation systems. We compare our system and Symbolab about the ability to solve problems and the meeting requirements of an IPS in education.

*Comparison of the ability to solve problems:* The exercises are collected from books [32], [33]. They include three kinds:

(1) Problems about Matrices – Vectors.
(2) Problems about the Linear equations system.
(3) Problems about Vector spaces.

Before solving the problems, they have to be represented. The problems are specified based on the structure of the knowledge-based model. Rela-Ops model has the specification of facts, and it can represent many kinds of real facts. By using the Rela-Ops model, our program can represent the objects and facts of the problems more naturally, so the hypothesis of the problems is specified more appropriate for reality. The knowledge base in Symbolab is organized as frames, so it only can solve the kinds of exercises that were set up. Table 4 shows the comparison of some kinds of problems that can be solved by programs.

*Example 17:* Compute $A + B^2$ with:

$$A = \begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & -36 \\ -36 & -31 & -21 \end{pmatrix} \quad B = \begin{pmatrix} -49 & 40 & -9 \\ -26 & 35 & 6 \\ -7 & -31 & -37 \end{pmatrix}$$

*The Solution of Our System:*
Step 1: We have:

$$B^2 = \begin{pmatrix} -49 & 40 & -9 \\ -26 & 35 & 6 \\ -7 & -31 & -37 \end{pmatrix}$$

$$= \begin{pmatrix} 1424 & -281 & 1014 \\ 322 & -1 & 222 \\ 1408 & -281 & 1246 \end{pmatrix}$$

Step 2:

$A + B^2$

$$= \begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & -36 \\ -36 & -31 & -21 \end{pmatrix} + \begin{pmatrix} 1424 & -281 & 1014 \\ 322 & -1 & 222 \\ 1408 & -281 & 1246 \end{pmatrix}$$

$$= \begin{pmatrix} -5+1424 & 1-281 & 34+1014 \\ 14+322 & -15-1 & -36+222 \\ -36+1408 & -31-281 & -21+1246 \end{pmatrix}$$

$$= \begin{pmatrix} 1419 & -280 & 1048 \\ 336 & -16 & 258 \\ 1372 & -249 & 1225 \end{pmatrix}$$

*The Solution of Symbolab:*

$$\begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & 36 \\ -36 & -31 & -21 \end{pmatrix} + \begin{pmatrix} -49 & 40 & -9 \\ -26 & 35 & 6 \\ -7 & -31 & -37 \end{pmatrix}^2$$

$$= \begin{pmatrix} 1419 & -280 & 1048 \\ 336 & -16 & 258 \\ 1372 & -249 & 1225 \end{pmatrix}$$

Steps

$$\begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & 36 \\ -36 & -31 & -21 \end{pmatrix} + \begin{pmatrix} -49 & 40 & -9 \\ -26 & 35 & 6 \\ -7 & -31 & -37 \end{pmatrix}^2$$

$$\begin{pmatrix} -49 & 40 & -9 \\ -26 & 35 & 6 \\ -7 & -31 & -37 \end{pmatrix}^2 = \begin{pmatrix} 1424 & -281 & 1014 \\ 322 & -1 & 222 \\ 1408 & -218 & 1246 \end{pmatrix}$$

$$= \begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & 36 \\ -36 & -31 & -21 \end{pmatrix} + \begin{pmatrix} 1424 & -281 & 1014 \\ 322 & -1 & 222 \\ 1408 & -218 & 1246 \end{pmatrix}$$

$$\begin{pmatrix} -5 & 1 & 34 \\ 14 & -15 & 36 \\ -36 & -31 & -21 \end{pmatrix} + \begin{pmatrix} 1424 & -281 & 1014 \\ 322 & -1 & 222 \\ 1408 & -218 & 1246 \end{pmatrix}$$

$$= \begin{pmatrix} 1419 & -280 & 1048 \\ 336 & -16 & 258 \\ 1372 & -249 & 1225 \end{pmatrix}$$

$$= \begin{pmatrix} 1419 & -280 & 1048 \\ 336 & -16 & 258 \\ 1372 & -249 & 1225 \end{pmatrix}$$

*Example 18:* Find the based-set of the vector space V from its spanning-set of vectors:

$$u_1 = [1, 2, -3], \quad u_2 = [-4, 5, 6],$$
$$u_3 = [7, 8, -9], \quad u_4 = [10, -11, 12]$$

With this problem, Symbolab cannot solve problems of the knowledge domain about vector spaces. Here is the solution to our system.

+ The solution of this system:
The solution to this problem is found by using algorithm 4. The current problem is in the knowledge domain $\mathcal{K}_3$ (Vector Space). Firstly, using the rule "*The set of roots of a linear equations system with n variables is a vector subspace of* $\mathbb{R}$", the problem will be changed to the problem about solving a linear equations system

$$\{x + 2y - 3z = 0, -4x + 5y + 6z = 0,$$
$$7x + 8y - 9z = 0, 10x - 11y + 12z = 0\}$$

This problem is in the knowledge domain $\mathcal{K}_2$ (Linear equations systems). Secondly, using the rule for *transforming the set of roots to the augmented matrix*, the problem in $\mathcal{K}_2$ will be changed to the problem of finding the augmented matrix. This problem is in the knowledge domain $\mathcal{K}_1$ (Matrices). It can be solved easily by using the transformations on a matrix. Finally, the solution of the original problem will be found by computing inversely.

*Step 1:* From $\{u_1, u_2, u_3, u_4\}$ is a spanning-set of V, we have:

$$A = \begin{bmatrix} 1 & 2 & -3 \\ -4 & 5 & 6 \\ 7 & 8 & -9 \\ 10 & -11 & 12 \end{bmatrix}$$

*Step 2:* Transform A to an echelon matrix:

$$A = \begin{bmatrix} 1 & 2 & -3 \\ -4 & 5 & 6 \\ 7 & 8 & -9 \\ 10 & -11 & 12 \end{bmatrix}$$

$$\begin{matrix} (2) \leftarrow (2)+4.(1) \\ (3) \leftarrow (3)-7.(1) \\ (4) \leftarrow (4)-10.(1) \end{matrix} \begin{bmatrix} 1 & 2 & -3 \\ 0 & 13 & -6 \\ 0 & -6 & 12 \\ 0 & -31 & 42 \end{bmatrix}$$

$$\begin{matrix} (2) \leftarrow 1/13.(2) \\ (3) \leftarrow 1/6.(3) \end{matrix} \begin{bmatrix} 1 & 2 & -3 \\ 0 & 1 & -\frac{6}{13} \\ 0 & -1 & 2 \\ 0 & 31 & 42 \end{bmatrix}$$

$$\begin{matrix} (3) \leftarrow (3)+(2) \\ (4) \leftarrow (4)-31.(2) \end{matrix} \begin{bmatrix} 1 & 2 & -3 \\ 0 & 1 & -\frac{6}{13} \\ 0 & 0 & \frac{20}{13} \\ 0 & 0 & \frac{719}{13} \end{bmatrix}$$

$$\begin{matrix} (3) \leftarrow 13/20.(3) \\ (4) \leftarrow (4)-719/13.(3) \end{matrix} \begin{bmatrix} 1 & 2 & -3 \\ 0 & 1 & -\frac{6}{13} \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

*Step 3:* This matrix has 03 non-zero rows:

$$[1, 2, -3]$$
$$[0, 1, -6/13]$$
$$[0, 0, 1]$$

**TABLE 5.** The results of testing the problems in [33].

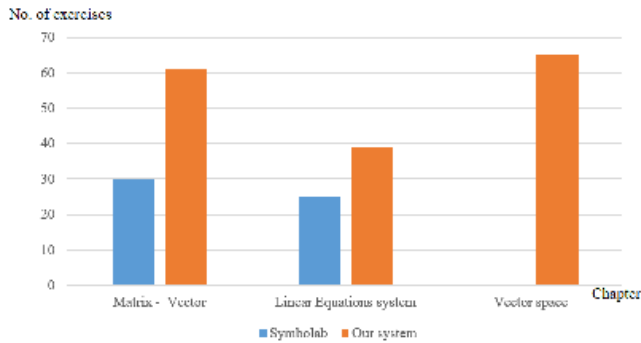| Chapter | Testing problems | Problems can be solved | |
|---|---|---|---|
| | | Symbolab | Our system |
| Matrix - Vector | 65 | 30 | 61 |
| Linear Equations system | 39 | 25 | 39 |
| Vector space | 67 | 0 | 65 |
| **Total** | **171** | **55** | **165** |



**FIGURE 5.** Results of testing in linear algebra course.

*Step 4:* Let

$$B := \{v_1 = [1, 2, -3],$$
$$v_2 = [0, 1, -6/13],$$
$$v_3 = [0, 0, 1]\}$$

Then B is a based-set of V.

The results of our system and Symbolab for solving problems in [33] are shown in Table 5 and Fig. 5:

The knowledge base of Symbolab is designed as frames, so it only can solve the kinds of exercises that had been set up, especially it cannot solve exercises in a section about vector spaces. The knowledge base of our system is organized as a complete system. It can represent the knowledge liking the knowledge acquisition of the human, so it can solve some advanced problems. However, some exercises are too hard for the typical students, and they require the depth knowledge in linear algebra to solve them, such as combining the knowledge of calculus to compute the determinant of a general matrix; hence our system cannot solve it.

*Comparison of the requirements of IPS in education:* Beside solving the problems in linear algebra, the system has to support students to study this course, so it should meet the requirements of an IPS in education. Table 6 compares systems based on these criteria.

## B. DESIGN AN INTELLIGENT SYSTEM FOR SOLVING PROBLEMS IN 2D-ANALYTIC GEOMETRY

2D-Analytic Geometry is an important knowledge domain in the high-school mathematical curriculum of Vietnam [34]. In this course, the student has to solve geometric plane

**TABLE 6.** Comparison based on requirements of an IPS.

| Requirements | Symbolab | Our system |
|---|---|---|
| *Sufficient knowledge-base* | • The knowledge base is sufficient and adequate for chapters about Matrices and Linear equations system. | |
| | • The knowledge base in Symbolab is organized as frames, so it is localized. <br> • The knowledge about vector spaces is not represented in the knowledge base. | • The knowledge base is organized based on the curriculum of the course. The knowledge model is partitioned into the knowledge sub-domains. This representation is close to reality, simulates the understanding of the learner about linear algebra. <br> • The knowledge base is a complete system with chapters: Matrices and Linear equation system, and Vector spaces. These sub-knowledge domains are combined with reasoning for solving a problem. |
| *Solve the common kinds of exercises in the course.* | • Solve conventional exercises in chapters about Matrices and Linear equations system. | |
| | • It only can solve kinds of problems that were designed as frames. <br> • It cannot solve problems with parameters. <br> • It cannot solve problems in the chapter about vector spaces. | • The program can solve the general problem. It is also able to solve exercises that require combining knowledge domains. <br> • It can solve some kinds of problems with parameters. <br> • It can solve some basic kinds of problems in the chapter about vector spaces. |
| *Pedagogy* | • Reasoning uses knowledge in linear algebra to solve the problem. <br> • The solutions are clear, step-by-step. | |
| | • The program is not built based on the curriculum of the course, so it is not a complete system for supporting the curriculum of a course. <br> • The solution does not count toward the student in university. They are very "hard" and not suitable for learners. | • The program is a complete system to support the student in studying linear algebra. The knowledge base and reasoning of this system toward the understanding of the learner. <br> • The solution of the system uses the knowledge which is suitable for the learner's level. It simulates the reasoning of the student to solve a problem. |
| *Usefulness for studying* | • The time for solving problems is acceptable. | |
| | • The program is not adequate to support the studying because its method is machinery; it only shows the solution as set up and does not the real reason to solve problems. | • The program can support studying because the solving method uses knowledge similar to that of students. |

problems by using the method of coordinate. In this section, we build an IPS for 2D-Analytic Geometry in high-school. The knowledge base of this system is represented by Rela-Ops model:

$$(\mathbb{C}, \mathbb{R}, \text{OPS}, \text{RULES})$$

The detail of this knowledge base is represented in Appendix B.

### 1) DESIGN THE INFERENCE ENGINE OF THE SYSTEM

For making inference processing of the system is faster and more effective, some heuristic rules are integrated into the inference engine for searching the proof of problems.

#### a: USE SAMPLE PROBLEMS

When solving a practical problem, we will consider a problem related to the current problem. If we had met the related problem, then we can use its results to get the solution to the practical problem more effectively. Sample Problems are related problems [35].

*Example 19:* Some sample problems have been used in our system:

*(Problem SP1):* Determining the equation of a line through two given points.

Sample problem SP1 $=$ (O, F) $\rightarrow$ G, in which:

$$O = \{A: \text{Point}, B: \text{Point}, d: \text{Line}\}$$
$$F = \{A = [x_A, y_A], B = [x_B, y_B],$$
$$A \text{ belongs } d, B \text{ belongs } d\}$$
$$G = \{\text{``Determine''}: d.\text{equation}\}$$

The solution of Sample problem:

Step 1: A: Point, B: Point,
  A belongs $d$, B belongs $d$,
  A $= [x_A, y_A]$, B $= [x_B, y_B]$,
  $\rightarrow d.\text{nvector} = [y_A - y_B, x_B - x_A]$

Step 2: A: Point, $n$: Vector
  $n$ is a normal vector of $d$, A belongs $d$,
  $d.\text{nvector} = [y_A - y_B, x_B - x_A]$, A $= [x_A, y_A]$
  $\rightarrow d.\text{equation}$
 $= ((y_A - y_B).(\_x - x_A) + (x_B - x_A).(\_y - y_A) = 0)$
 $= ((y_A - y_B).\_x + (x_B - x_A).\_y - x_A.(y_A - y_B) - y_A.$
$(x_B - x_A) = 0)$

*(Problem SP2):* Determining the equation of a line through a given points and perpendicular with another given line.

Sample problem SP2 $=$ (O, F) $\rightarrow$ G, in which:

$$O = \{A: \text{Point}, d: \text{Line}, f: \text{Line}\}$$
$$F = \{A = (x_A, y_A), f \text{ determined}, A \text{ belong } d, d \perp f\}$$
$$G = \{\text{``Determine''}: d.\text{equation}\}$$

Solution of Sample problem:

Step 1: $\{f \text{ determined}\} \rightarrow \{f.\text{equation}\}$
Step 2: $\{f. \text{equation}\} \rightarrow \{f.\text{nvector} = [f_1, f_2]\}$
Step 3: $\{f. \text{nvector} = [f_1, f_2], d \perp f\} \rightarrow \{d.\text{nvector} = [-f_1, f_2]\}$
Step 4: $\{A = [x_A, y_A], d.\text{nvector} = [-f_1, f_2], A \text{ belongs } d\}$
  $\rightarrow d.\text{equation} = (-f_1.(\_x - x_A) + f_2.(\_y - y_A) = 0)$
  $= (-f_1.\_x + f_2.\_y - f_1.x_A + f_2.y_A = 0)$

The algorithm for searching the sample problem in the knowledge domain about 2D-Analytic Geometry [35].

*Algorithm 5:* Give a problem P with the hypothesis **(O, F)** in the knowledge domain about 2D-Analytic Geometry. A sample problem for the problem P can be founded by these steps:

**Input:**
  **(O, F)**: hypothesis of the current problem P.
  *Sample*: a set of sample problems.
**Output:** Determine a problem S $\in$ *Sample* is a sample problem of P.

---
**Step 1:**
  Sample_found:= false;
  **for** S in *Sample* **do**
    **if** (S.O $\sqsubseteq$ **O** and S.F $\sqsubseteq$ **F**) **then**
      Sample_found:= true;
      break;      **end if;**
  **end do;**

**Step 2: if** (Sample_found) **then**
    **Return** S is a sample problem of P.
---

#### b: GROUP OF RULES CAN BE APPLIED

When solving a problem, there are some rules that have more possibilities to apply for this problem. These rules belong to objects and facts in the practical problem. Hence, based on the hypothesis of the problem, we can show a group of rules that have the possibility to be used for solving a problem.

*Example 20:* Some groups of rules:

- Group of rules for problems about determining the equation of a unique line in a triangle, such as a bisector line, median line, and the height of a triangle.
- Group of rules for problems about determining the distance between a point and a line in a circle.

### 2) TESTING AND EXPERIMENTAL RESULTS

Exercises of the course are collected from [34]. These exercises are classified into four kinds:

(i) Computing the coordinate of a point.
(ii) Determining an equation of a line.
(iii) Problems about a triangle, such as determining the equation of a line in a triangle (median line, bisector line, height line); determining a point in a triangle, computing the area, the radius of a circle.
(iv) Problems about computing the value of a parameter to satisfy some given conditions.

Symbolab only can solve some fundamental problems in kind (i) and kind (ii), such as: determining an intersection point of two lines, the equation of a line through two given points. Hence, it is not an intelligent system for supporting to learn this course. Our system satisfies the requirements of an IPS in education.

*Example 21:* Given a triangle ABC with B(0, 3) and the median line (CM): $4x + y + 1 = 0$. Let $d$ be the height line of triangle ABC, $d$ through point A, $(d)$: $-4x + 3y = 0$. Compute the area of triangle ABC.

+ Specification the problem:

O = {Triangle(A,B,C), M: Point, $d$: line}

Re = {line(C, M) is the median line of Triangle(A,B,C),

       $d$ is the height line of Triangle(A,B,C),

       A belong $d$}

E = {line(C, M).equation = ($4x + y + 1 = 0$),

       $d$.equation = ($-4x + 3y = 0$)}

G = {Compute: Triangle(A,B,C).area}

+ The solution of our system:

Using algorithm 1, the solution of this problem is found by followed reasoning steps:

S1 = {line(C, M) is the median line of Triangle(A,B,C)}
     $\rightarrow$ {M is midpoint of AB}

S2 = {$d$ is the height line of Triangle(A,B,C), A belong
     $d$} $\rightarrow$ {$d \perp$ line(B, C)}

S3 = {C belong line(B, C), C belong line(C, M)} $\rightarrow$
     {C = line(B, C) intersection line(C, M)}

S4 = {d.equation} $\rightarrow$ {d.nvector}

S5 = {$d$.nvector, $d \perp$ line(B, C)} $\rightarrow$ {line(B,C).nvector}

S6 = {M midpoint AB, B} $\rightarrow$ {M.$x$ = 1/2 (A.$x$ + 3),
                     M.$y$ = 1/2 A.$y$}

S7 = {line(B, C).nvector, B, B belong line(B, C)}
       $\rightarrow$ {line(B, C).equation}

S8 = {C = line(B, C) intersection line(C, M),
     line(B, C).equation, line(C, M).equationn} $\rightarrow$ {C}

S9 = {A belong d, d.equation} $\rightarrow$ {$-4^{*}$A.$x$ + $3^{*}$A.$y$ = 0}

S10 = {M belong line(C, M), line(C, M).equation}
     $\rightarrow$ {$4^{*}$M.$x$ + M.$y$ + 1 = 0}

S11 = {$-4^{*}$A.$x$ + $3^{*}$A.$y$ = 0, $4^{*}$M.$x$ + M.$y$ + 1 = 0,
      M.$x$ = 1/2 (A.$x$ + 3), M.$y$ = 1/2 A.$y$} $\rightarrow$ {A.$x$, A.$y$}

S12 = {A.$x$, A.$y$} $\rightarrow$ {A}

S13 = {Triangle(A,B,C),A,B,C}$\rightarrow${Triangle(A,B,C).area}

In the above reasoning steps, some heuristic rules have been used in the inference process:

- The steps S1, S2, and S3 use the group of rules relating to individual lines in a triangle.
- The group of steps S5-S8 uses the Sample Problem SP2.

From those reasoning steps, the solution of this problem is as follows:

**Step 1:**
{line(C, M) is the median line of Triangle(A,B,C)}
$\rightarrow$ {M is midpoint of AB}

**Step 2:**
{$d$ is the height line of
        Triangle(A,B,C),
   A belong $d$}
$\rightarrow$ {$d \perp$ line(B, C)}

**Step 3:**
{C belong line(B, C),
C belong line(C, M) }
$\rightarrow$ {C = line(B, C) intersection
          line(C, M)}

**Step 4:**
{d.equation = ($-4x + 3y = 0$)}
$\rightarrow$ {d.nvector = [$-4$, 3]}

**Step 5:**
{$d$.nvector = [$-4$, 3],
$d \perp$ line(B, C)}
$\rightarrow$ {line(B, C).nvector
           = [$-3$, $-4$]}

**Step 6:**
{M midpoint AB,
B = [0, 3]}
$\rightarrow$ {M.$x$ = 1/2 (A.$x$ + 3),
     M.$y$ = 1/2 A.$y$}

**Step 7:**
{line(B, C).nvector = [$-3$, $-4$],
B = [3, 0],
B belong line(B, C) }
$\rightarrow$ {line(B, C).equation
     = ($-3x - 4y + 9 = 0$)}

**Step 8:**
{C = line(B, C) intersection line(C, M),
line(B, C).equation
     = ($-3x - 4y + 9 = 0$),
line(C, M).equation
     = ($4x + y + 1 = 0$)}
$\rightarrow$ {C = [$-1$, 3]}

**Step 9:**
{A belong d,
d.equation
        = ($-4x + 3y = 0$)}
$\rightarrow$ {$-4^{*}$A.$x$ + $3^{*}$A.$y$ = 0}

**Step 10:**
{M belong line(C, M),
line(C, M).equation
     = ($4x + y + 1 = 0$)}
$\rightarrow$ {$4^{*}$M.$x$ + M.$y$ + 1 = 0}

**Step 11:**
{$-4^{*}$A.$x$ + $3^{*}$A.$y$ = 0,
$4^{*}$M.$x$ + M.$y$ + 1 = 0,

**TABLE 7.** The results of testing the problems in [34].

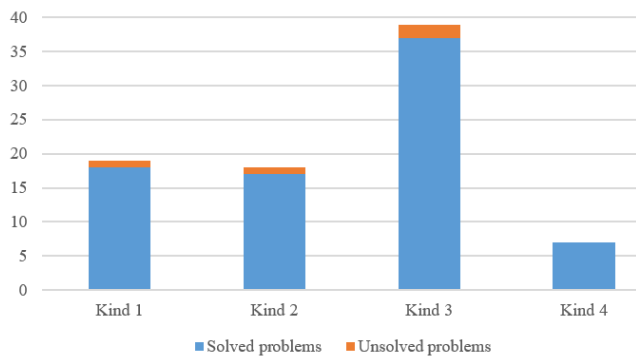| Order | Kind | Testing problems | Solved problems by our system |
|---|---|---|---|
| 1 | Computing the coordinate of a point | 19 | 18 |
| 2 | Determining an equation of a line | 18 | 17 |
| 3 | Problems about a triangle | 39 | 37 |
| 4 | Problems about computing the value of a parameter | 5 | 5 |
| | Total | 83 | 79 |

**TABLE 8.** Results of the survey.

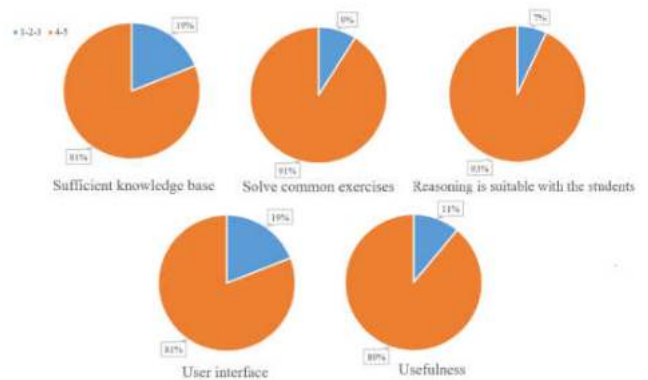| Criterion | Level (Very bad → Very good) | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| A knowledge-base is sufficient. | | 19% | | 81% | |
| The program can solve common exercises. | | 9% | | 91% | |
| Program's user interface is friendly, easy to use. | | 19% | | 81% | |
| Pedagogy: the solution and its reasoning are suitable with the students. | | 7% | | 93% | |
| The program is useful to support studying. | | 11% | | 89% | |



**FIGURE 6.** Results of testing in a 2D-Analytical Geometry course.



**FIGURE 7.** Results of the survey in a 2D-Analytical Geometry course.

M.$x$ = 1/2 (A.$x$+ 3),
M.$y$ = 1/2 A.$y$
→ {A.$x$ = -21/8,
    A.$y$ = −7/2}

**Step 12:**
{A.$x$ = −21/8, A.$y$ = −7/2}
→ A = [−21/8, −7/2]

**Step 13:**
{Triangle(A, B, C),
A = [−21/8, −7/2],
B = [0, 3], C = [−1, 3]}
→ Triangle(A,B,C).area
        = 247/16

This solution is similar to the reasoning of the student for solving this problem. It uses the knowledge of 2D-Analytic Geometry in high-school. Using heuristics rules makes the inference process is more efficient in practice.

Table 7 and Fig. 6 show the ability of our IPS system for solving problems in this course:

The knowledge base of our system is organized as a complete system. It can solve many kinds of exercises in these courses. More than that, our system is pedagogical and suitable for the knowledge level of high-school students. This system is helpful in supporting the students in studying this course.

Our program has been tested and evaluated by 100 students of two high-schools in Ho Chi Minh City, Vietnam. All students had just graduated from high school in August 2018. Those students include 11 average students who have a GPA from 5.0 – 7.0, 55 good students who have a GPA from 7.0 – 8.0, and 34 outstanding students who have a GPA higher than 8.0.

This survey is interested in the requirements of an IPS in education: the sufficient of a knowledge base, the effectiveness of problem-solving, the pedagogy of the program, and the usefulness for studying this course. Firstly, each student selects 03 exercises from a set of solvable exercises (79 exercises). He/She checks their solutions. Secondly, the student inputs two other problems in four kinds of exercises. The program solves and shows solutions for those exercises. Finally, they assess each criterion with a level from 1 – 5. Each level has the meaning as in [38], it is respectively very bad – very good.

The results of this survey are shown in Table 8 and Fig.7:

Though the results of the survey, our program meets the requirements of IPS in education. It is also received excellent feedback from students. Our program is useful to support students studying this course. Its solutions are suitable for the knowledge level of students. The reasoning uses rules, theorems in the curriculum of this course. However, this

**TABLE 9.** Methods for knowledge representation in IPS.

| Method | Universality | Usability | Practicality | Formality |
|---|---|---|---|---|
| Ontology | Level 3 | Level 3 | Level 2 | Level 3 |
| COKB | Level 3 | Level 4 | Level 3 | Level 2 |
| Frame | Level 2 | Level 2 | Level 2 | Level 2 |
| Algebraic approach | Level 1 | Level 2 | Level 1 | Level 4 |
| Rela-Ops model | Level 4 | Level 4 | Level 4 | Level 3 |

program needs more research about its impact on developing students' solving-problem skills.

## VII. CONCLUSION

In this paper, we propose a method to represent the Rela-Ops model, which combines the knowledge of relations and operators. This model includes components: concepts, relations, operators, and rules. The syntax of sentences and the unification of facts have also been defined. They make this model remaining flexible and more effective in practice. Based on the structure of the model, the problems have also been proposed and solved. Reasoning processing uses the characteristics of relations and operators to solve these problems. The algorithms have been proved the effectiveness

Our proposed method can apply in many knowledge domains of STEM education. This model and some its restriction model were applied to build IPS systems in another mathematics courses, such as knowledge domains about Plane Geometry in middle school [11], Algebra in middle school [37], Solid Geometry in high school [9], Vector Algebra in high school [10], Discrete Mathematics in university [38]. In this paper, we presented the application of the Rela-Ops model in designing an IPS for 2D-Analytic Geometry in high-school. Table 9 compares the methods of knowledge representation based on the criteria of an IPS system in education.

Besides that, a knowledge model for multiple knowledge domains is studied, in which each sub-domain has the form as the Rela-Ops model. This model represents the connective relations between these domains. Besides, the problems for combining the knowledge of these domains are proposed and solved. In using this model, the knowledge base of linear algebra has been represented. This knowledge domain includes three domains: Matrices, Linear equations systems, and Vector spaces. This knowledge base is applied to design an intelligent system for solving problems in linear algebra. Its proofs are readable and human-alike solutions. This system can be used in supporting the learning of students.

In the future, problems with optimization of the searching rules on the Rela-Ops model will be studied. Some problems with the knowledge model for multiple knowledge domains will also be researched, such as the problems about the integration of knowledge domains for solving current problems. Our methods would be proved the merits when they could

be applied in building the efficient, intelligent systems which have the knowledge from multiple domains, such as knowledge domains about mathematics, chemistry, and physics.

## APPENDIX
### A. KNOWLEDGE BASE OF LINEAR ALGEBRA COURSE IN THE UNIVERSITY

The knowledge domain about Linear Algebra is partitioned into three knowledge sub-domains.

$$\mathbb{K} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$$

$\mathcal{K}_1$ – Knowledge domain about Matrices

$$\mathcal{K}_1 = (\mathbb{C}_1, \mathbb{R}_1, \mathbb{OPS}_1, \mathbb{RULES}_1)$$

$\mathbb{C}_1$ – set of concepts about the matrix, vector, and their types.

$$\mathbb{C}_1 = \{\text{MATRIX, SQUARE\_MATRIX,}$$
$$\text{DIAGONAL\_MATRIX, VECTOR, } \ldots\}$$

The structures of MATRIX concept and SQUARE_MATRIX are as example 2.1

$\mathbb{R}_1$– set of relations between the concepts in $\mathbb{C}_1$

$$\mathbb{R}_1 = \{equal, \ row \ equivalence, \ column \ equivalence,$$
*eigenvalue, eigenvector*}

It also includes relations "is-a" between the kinds of matrixes, such as: SquareMatrix is-a Matrix, DiagnolMatrix is-a SquareMatrix...

$\mathbb{OPS}_1$ is referring to a set of operators between matrices and vectors concepts.

+ The operators: add (+), multiply between two matrixes

+ The operators: determinant (det), inverse $(^{-1})$ on a matrix.

+ The operators about row transformation and column transformation on a matrix.

+ The operators: add (+), inner product, outer product (o) between two vectors.

$\mathbb{RULES}_1$ – set of rules on the matrices and vectors

Rule 1.1: {A: Square_Matrix, $\exists \alpha \in \mathbb{R}$, $\exists m_1$, $\exists m_2$, $m_1 \neq m_2$, $\text{row}_A(m_1) = \alpha^* \text{row}_A(m_2)$}}
  $\rightarrow$ {A.det= 0}

Rule 1.2: {A: Square_Matrix, A.diag = 1}
  $\rightarrow \exists$D: Diagnol_Matrix, S: A: Square_Matrix,

  S.inv = 1, D.n = S.n = A.n: A = $S^{-1}$.D.S

Rule 1.3: A, B: Matrix, A.n = B.m:

$$(A.B)^T = B^T.A^T$$

Rule 1.4: {A: Matrix, B: Matrix}
A is *row equivalence* to B $\leftrightarrow \exists [f_1, \ldots f_n] \subset \mathbb{OPS}_1$: list of operators about primary row transformation,

$$f_n(\ldots(f_1(A))\ldots) = B$$

$\mathcal{K}_2$ – Knowledge domain about linear equations systems

$$\mathcal{K}_2 = (\mathbb{C}_2, \mathbb{R}_2, \mathbb{RULES}_2)$$

$\mathbb{C}_2$ is referring to a set of concepts about the linear equation and the *linear equations system.*

$$\mathbb{C}_2 = \{\text{EQUATION, EQUATIONS\_SYSTEM,}$$
$$\text{CRAMER\_SYSTEM}\}$$

*Example 1:* The structure of EQUATIONS_SYSTEM concept:

*Attrs*:= {m, n, eq[m], Root, aug_matrix}
  m: $\mathbb{N}$ // number of equations
  n: $\mathbb{N}$ // number of variables
  eq[m]: EQUATION // List of linear equations

$$\text{Root} := \left\{ (b_1, ...b_n) \in \mathbb{R}^n | \forall i = \overline{1, m} : \right.$$

$$\left. \sum_{j=1}^{n} pt[i].a[j]^* b_j = pt[i].a[n+1] \right\}$$

  aug_matrix: MATRIX [m, n+1] // Augmented matrix
  *Facts*:={$\forall$ i, $1 \le i \le m$, eq[i].n = n}
  *EqObj*:={$\forall$ i, j, $1 \le i \le m$, $1 \le j \le n+1$:
  aug_matrix[i, j] = eq[i].a[j] ...}
  *RulObj*:={$r_1$: aug_matrix.rank= n
                        $\rightarrow$ card(Root) = 1...}
$\mathbb{R}_2$ – set of relations between the concepts in $\mathbb{C}_2$

$$R_2 = \{equivalence\}$$

$\mathbb{RULES}_2$ – Set of rules on linear equations systems
Rule 2.1: A, B: LinearEquationSystem,

A is *equivalence* to B $\leftrightarrow$ A.Root = B.Root

Rule 2.2: A, B: LinearEquationSystem

A is *equivalence* to B
    $\leftrightarrow$ (A.aug_matrix is row equivalence
    to B.aug_matrix)

$\mathcal{K}_3$ – Knowledge domain about vector spaces

$$\mathcal{K}_3 = (C_3, \mathbb{R}_3, \mathbb{OPS}_3, \mathbb{RULES}_3)$$

$\mathbb{C}_3$ – Set of concepts on vector spaces

$$\mathbb{C}_3 = \{\text{VECTOR, VECTOR\_SPACE}\}$$

*Example 2:* The structure of VECTOR_SPACE concept:
  *Attrs*:= {dim, L}
    dim: $\mathbb{R}$ // dimension
    L $\subseteq$ I$_{\text{VECTOR}}$
  *Facts*:= Ø *EqObj*:= Ø
  *RulObj*:={$r_1$: {$\forall$u, v $\in$ L, $\forall$ k$\in \mathbb{R}$} $\rightarrow$ {ku + v $\in$ L}
            $r_2$: {$\forall$v $\in$ L} $\rightarrow$ $\exists$ v $\in$ L: u + v = 0} }
$\mathbb{R}_3$ – set of relations between concepts in $\mathbb{C}_3$
    $\mathbb{R}_3$ = {*belong, sub-space, based-set, spanning-set,*
            *linearly independent* ... }
*Example 3:*
+ *based-set* $\subseteq$ I$_{2^{I_{\text{VECTOR}}}}$ $\times$ I$_{\text{KHONGIGANVECTOR}}$: it means a set of vectors is a based-set of a vector space.

+ *linearly independent* $\in$ I$_{\text{VECTOR}}^{\text{k}}$: a relation about linearly independent between k vectors.
  $\mathbb{OPS}_3$ – set of operators between two vectors and two vector space.
*Example 4:* V: VectorSpace
  •*CoorMatrix*$_V$: $2^{I_{\text{Vector}}} \times 2^{I_{\text{Vector}}} \rightarrow$ I$_{\text{SquareMatrix}}$

$$(\text{B1, B2}) \mapsto \text{M}$$

*CoorMatrix*$_V$ is an operator to determine the matrix for convert the coordinate in a vector space V from based-set B1 to based-set B2.
  •*Coor*$_V$: I$_{\text{Vector}} \times 2^{I_{\text{Vector}}} \rightarrow$ I$_{\text{Vector}}$

$$(\text{v, B}) \mapsto \text{v}'$$

*Coor*$_V$ is an operator to determine the coordinate of a vector v with based-set B in a vector space V.
  $\mathbb{RULES}_3$ – set of rules on vector spaces
  Rule 3.1: {W, V: VectorSpace}

W is a *sub-space* of V $\leftrightarrow$ W.L $\subset$ V.L

Rule 3.2: {S: $2^{I_{\text{VECTOR}}}$, S = {$e_1, e_2, ..., e_k$} }

S is linear independence
    $\leftrightarrow$ ({$a_1e_1+ ...+a_ke_k = 0$}
    $\leftrightarrow$ {$a_1 = a_2 = ... = a_m = 0$})

Rule 3.3: {V: VectorSpace, B: $2^{I_{\text{VECTOR}}}$, B = {$e_1$, $e_2, ..., e_{V.dim}$}}

B is a *based-set* of V
    $\leftrightarrow$ (B is linearly independent) AND
    (B is a spanning-set of V)

Rule 3.4: {V: VectorSpace, *v*: VECTOR, B1, B2: $2^{I_{\text{VECTOR}}}$, B1 is a based-set of V, B2 is a based-set of V}

Coor$_V$(*v*, B2) = CoorMatrix$_V$(B2, B1). Coor$_V$(*v*, B1)

### B. KNOWLEDGE BASE OF 2D-ANALYTIC GEOMETRY COURSE IN THE HIGH-SCHOOL
This knowledge domain is presented by Rela-Ops model:

$$\mathcal{K} = (\mathbb{C}, \mathbb{R}, \mathbb{OPS}, \mathbb{RULES})$$

$\mathbb{C}$ – set of concepts in the knowledge domain:

$$C = \{POINT, LINE, VECTOR, TRIANGLE, CIRCLE\}$$

*Example 5:*
• The structures of VECTOR concept:
  *Attrs* = {x, y, module}
    x, y: $\mathbb{R}$ // abscissa and ordinate of a vector
    module: $\mathbb{R}$ // module of a vector
  *Facts*:= Ø
  *EqObj*:= {module = $\sqrt{x^2 + y^2}$}
  *RulObj*:= Ø
• The structures of LINE concept:
  *Attrs* = {equation, nvector, _x, _y}

_x, _y: ℝ // parameters of a line

nvector: VECTOR // normal vector of a line.

equation: expression // equation of a line with parameters _x and _y

*Facts*:= Ø

*EqObj*:= {equation =
$$(nvector.x^*\_x + nvector.y^*\_y + c = 0)$$}

*RulObj*:= Ø

ℝ – set of relations:

*Example 6:* Some of the relations in this knowledge domain:

● Relations between a point and a line or a segment: a point *belongs* to a line (or a segment), a point is a *midpoint* of a segment.

● Relations between a line and a triangle: a line is a *bisector* (or *height*, or *median*) line of a triangle.

● Relations between two vectors: *parallel (//)*, *perpendicular (⊥)*.

● Relations between two lines: *parallel (//)*, *perpendicular (⊥)*, *intersection*

● Relations between a vector and a line: a vector is a *normal* (or *direction*) vector of a line.

● It also includes relations *"is-a"* between the kinds of triangles, such as Right Triangle is-a Triangle, Isosceles Triangle is-a Triangle, Equilateral Triangle is-a Isosceles Triangle. . .

OPS is referring to a set of operators between vectors.

| Operator | Meaning | Arguments | Return | Properties |
|---|---|---|---|---|
| + | Sum of vectors | Vector × Vector | Vector | commutative associative identity |
| - | Difference between two vectors | Vector × Vector | Vector | |
| * | Product between a real number and a vector | ℝ × Vector | Vector | |
| . | Inner product of vectors | Vector × Vector | ℝ | Commutative |
| O | Cross product of vectors | Vector × Vector | Vector | |

RULES – set of inference rules

+ Some deductive rules in RULES-set:

R1: {a,b,c: vector, c = a o b} → {c ⊥ a, c ⊥ b}

R2: {P, Q, M: Point, M midpoint PQ}
$$→ \{M = [1/2.(P.x + Q.x), 1/2.(P.y + Q.y)]\}$$

R3: {A: Point, B: Point, d: line,
A belong d, B belong d}
$$→\{d.nvector.vector(AB)= 0\}$$

+ Some rules for generating an object:

R5: {ABC: triangle, d: line, A belong d, d is the height line of ABC}
$$→ \{H: Point, H = d \text{ intersection } BC\}$$

R6: {ABC: triangle, d: line, A belong d, d is the median line of ABC}
$$→ \{M: Point, M \text{ midpoint } BC, M \text{ belong } d\}$$

+ Some equivalent rules in RULES-set:

R7: {a: vector, b: vector, k: ℝ, $k \neq 0$}
$$\{a = k^* b\} ↔ \{a.x = k^*b.x, a.y = k^*b.y\}$$

R8: {u: vector, d: line}
$$\{u ⊥ d\} ↔ \{u \text{ // } d.nvector\}$$

R9: {d1: line, d2: line}
$$\{d1 \text{ // } d2\} ↔ \{d1.nvector = d2.nvector\}$$

R10: {a, b: vector}
$$\{a.b = 0\} ↔ \{a ⊥ b\}$$

+ Some equation rules in RULES-set:

R11: P, Q: Point, vector(PQ) = [Q.x – P.x, Q.y – P.y]

R12: P, Q: Point,
$$PQ = \sqrt{(P.x - Q.x)^2 + (P.y - Q.y)^2}$$

R13: a, b: vector, a.b = a.x*b.x + a.y*b.y

R14: a, b: vector, a o b = - (a o b)

## REFERENCES

[1] N. Noy and D. McGuinness, Eds., "Final report on the 2013 NSF workshop on research challenges and opportunities in knowledge representation," in *Proc. Nat. Sci. Found. Workshop Rep.*, Aug. 2013, pp. 44–49.

[2] C. LeGassick *et al.*, "Artificial intelligence index—2017 annual report," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2017. Accessed: Mar. 30, 2020. [Online]. Available: https://hai.stanford.edu/ai-index-2017

[3] B. D. Nye, P. I. Pavlik, A. Windsor, A. M. Olney, M. Hajeer, and X. Hu, "SKOPE-IT (shareable knowledge objects as portable intelligent tutors): Overlaying natural language tutoring on an adaptive learning system for mathematics," *Int. J. STEM Educ.*, vol. 5, no. 1, p. 12, Dec. 2018, doi: 10.1186/s40594-018-0109-4.

[4] I. Hatzilygeroudis and J. Prentzas, "Knowledge representation requirements for intelligent tutoring systems," in *Proc. 7th Int. Conf. Intell. Tutoring Syst. (ITS)*, in Lecture Notes in Computer Science, vol. 3220. Berlin, Germany: Springer, 2004, pp. 87–97.

[5] N. Van Do, "Intelligent problem solvers in education: Design method and applications," in *Intelligent Systems*, V. M. Koleshko, Ed. Rijeka, Croatia: InTech, 2012, ch. 6, pp. 121–148.

[6] H. D. Nguyen, N. V. Do, N. P. Tran, and X. H. Pha, "Criteria of a knowledge model for an intelligent problems solver in education," in *Proc. IEEE Int. Conf. Knowl. Syst. Eng. (KSE)*, Ho Chi Minh, Vietnam, Nov. 2018, pp. 288–293.

[7] R. Davis, H. Shrobe, and P. Szolovits, "What is knowledge representation?" *AI Mag.*, vol. 14, no. 1, pp. 17–33, 1993.

[8] R. J. Brachman and H. Levesque, *Knowledge Representation and Reasoning*. San Mateo, CA, USA: Morgan Kaufmann, 2004.

[9] N. V. Do, H. D. Nguyen, and A. Selamat, "Knowledge-based model of expert systems using rela-model," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, no. 8, pp. 1047–1090, Aug. 2018.

[10] K. Inoue, N. V. Do, V. T. Pham, and H. D. Nguyen, "Solving problems on a knowledge model of operators and application," *Int. J. Digit. Enterprise Technol.*, vol. 1, nos. 1–2, pp. 37–59, 2018.

[11] D. H. Nguyen, T. V. Pham, T. T. Le, and H. D. Tran, "A mathematical approach for representation knowledge about relations and its application," in *Proc. IEEE Int. Conf. Knowl. Syst. Eng. (KSE)*, Ho Chi Minh, Vietnam, Oct. 2015, pp. 324–327.

[12] H. D. Nguyen and N. V. Do, "Method for solving problems on a knowledge model of operators," in *Proc. IEEE Conf. e-Learn., e-Manage. e-Services (ICe)*, Langkawi, Malaysia, Oct. 2016, pp. 122–127.

[13] W. Gan and X. Yu, "Automatic understanding and formalization of natural language geometry problems using syntax-semantics models," *Int. J. Innov. Comput., Inf. Control*, vol. 14, no. 1, pp. 83–98, 2018.

[14] A. Y. Turhan, "Introductions to description logics—A guided tour," in *Proc. 9th Int. Summer School Reasoning Web Semantic Technol. Intell. Data Access*, in Lecture Notes in Computer Science, vol. 8067. Berlin, Germany: Springer, 2013, pp. 150–161.

[15] E. Grefenstette, "Towards a formal distributional semantics: Simulating logical calculi with tensors," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics (SEM)*, Atlanta, GA, USA, vol. 1, 2013, pp. 1–10.

[16] D. H. Nguyen, C. Sakama, T. Sato, and K. Inoue, "Computing logic programming semantics in linear algebra," in *Proc. 12th Multi-Disciplinary Int. Conf. Artif. Intell. (MIWAI)*, in Lecture Notes in Artificial Intelligence, vol. 11248. Hanoi, Vietnam: Springer, Nov. 2018, pp. 32–48.

[17] F. Kardoš, J. Pócs, and J. Pócsová, "On concept reduction based on some graph properties," *Knowl.-Based Syst.*, vol. 93, pp. 67–74, Feb. 2016.

[18] S. Nurdiati and C. Hoede, "25 years development of knowledge graph theory: The results and the challenge," *Memorandum*, vol. 1876, pp. 1–10, Sep. 2008.

[19] N. Guan, D. Song, and L. Liao, "Knowledge graph embedding with concepts," *Knowl.-Based Syst.*, vol. 164, pp. 38–44, Jan. 2019.

[20] W. Liu, J. Liu, M. Wu, S. Abbas, W. Hu, B. Wei, and Q. Zheng, "Representation learning over multiple knowledge graphs for knowledge graphs alignment," *Neurocomputing*, vol. 320, pp. 12–24, Dec. 2018.

[21] A. Wright and D. F. Sittig, "A framework and model for evaluating clinical decision support architectures," *J. Biomed. Inform.*, vol. 41, no. 6, pp. 982–990, Dec. 2008.

[22] X. Yu, M. Wang, W. Gan, B. He, and N. Ye, "A framework for solving explicit arithmetic word problems and proving plane geometry theorems," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 33, no. 7, Jun. 2019, Art. no. 1940005.

[23] W. Gan, X. Yu, T. Zhang, and M. Wang, "Automatically proving plane geometry theorems stated by text and diagram," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 33, no. 7, Jun. 2019, Art. no. 1940003.

[24] H. Wang, "Research on the model of knowledge representation ontology based on framework in intelligent learning system," in *Proc. Int. Conf. Electr. Control Eng.*, Yichang, China, Sep. 2011, pp. 6757–6760.

[25] R. Gayathri and V. Uma, "Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey," *ICT Express*, vol. 4, no. 2, pp. 69–74, Jun. 2018.

[26] V. N. Do and T. T. Mai, "Perfect COKB model and reasoning methods for the design of intelligent problem solvers," in *Proc. 16th Int. Conf. Intell. Softw. Methodol., Tools, Techn. (SOMET)*, in Frontiers in Artificial Intelligence and Applications, Kitakyushu, Japan, vol. 297, Sep. 2017, pp. 35–50.

[27] J. Calmet and A. I. Tjandra, "Building bridges between knowledge representation and algebraic specification," in *Methodologies for Intelligent Systems* (Lecture Notes in Computer Science), vol. 869. Berlin, Germany: Springer, 1994, pp. 295–304.

[28] T. Plotkin and M. Knyazhansky, "Symmetries of knowledge bases," *Ann. Math. Artif. Intell.*, vol. 64, no. 4, pp. 369–383, Apr. 2012.

[29] C. Sakama, H. D. Nguyen, T. Sato, and K. Inoue, "Partial evaluation of logic programs in vector spaces," in *Proc. 11th Workshop Answer Set Program. Other Comput. Paradigms (ASPOCP), Co-Located with Federated Logic Conf. (FLoC)*, Oxford, U.K., Jul. 2018, pp. 1–15.

[30] Y. Wang and O. A. Zatarain, "Design and implementation of a knowledge base for machine knowledge learning," in *Proc. IEEE 17th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI CC)*, Berkeley, CA, USA, Jul. 2018, pp. 70–77.

[31] M. Valipour and Y. Wang, "Formal properties and mathematical rules of concept algebra for cognitive machine learning," *J. Adv. Math. Appl.*, vol. 5, no. 1, pp. 69–86, Jun. 2016.

[32] S. Lang, "Linear algebra," in *Undergraduate Texts in Mathematics*, 3rd ed. Springer, 1987. [Online]. Available: https://www.springer.com/gp/book/9780387964126

[33] H. Anton and C. Rorres, *Elementary Linear Algebra*, 10th ed. Hoboken, NJ, USA: Wiley, 2010.

[34] *Textbook and Workbook of High School Mathematics*, Ministry Educ. Training, Hanoi, Vietnam, 2013.

[35] N. V. Do, H. D. Nguyen, and T. T. Mai, "Designing an intelligent problems solving system based on knowledge about sample problems," in *Proc. 5th Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, in Lecture Notes in Artificial Intelligence, vol. 7802. Kuala Lumpur, Malaysia: Springer, 2013, pp. 465–475.

[36] R. Anwar, M. Rehman, K. S. Wang, M. A. Hashmani, and A. Shamim, "Investigation of knowledge sharing behavior in global software development organizations using social cognitive theory," *IEEE Access*, vol. 7, pp. 71286–71298, 2019.

[37] N. Do and H. Nguyen, "A reasoning method on computational network and its applications," in *Proc. Int. MultiConf. Eng. Comput. Sci. (IMECS)*, Hong Kong, Mar. 2011, pp. 137–141.

[38] H. D. Nguyen and N. V. Do, "Intelligent problems solver in education for discrete mathematics," in *Proc. 16th Int. Conf. Intell. Softw. Methodol., Tools, Techn. (SOMET)*, in Frontiers in Artificial Intelligence and Applications, Kitakyushu, Japan, vol. 297, Sep. 2017, pp. 21–34.

[39] *Symbolab*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.symbolab.com

[40] *Mathway*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.mathway.com/

[41] *Wolfram|Alpha*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.wolframalpha.com/

[42] *Maple*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.maplesoft.com/support/install/maple2017_install.html

[43] *Matlab*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.mathworks.com/academia/student_version.html

[44] Y. Zhai, D. Li, and J. Zhang, "Variable decision knowledge representation: A logical description," *J. Comput. Sci.*, vol. 25, pp. 161–169, Mar. 2018.

[45] M. Chiachío, J. Chiachío, D. Prescott, and J. Andrews, "A new paradigm for uncertain knowledge representation by plausible Petri nets," *Inf. Sci.*, vol. 453, pp. 323–345, Jul. 2018.

[46] H.-C. Liu, X. Luan, Z. Li, and J. Wu, "Linguistic Petri nets based on cloud model theory for knowledge representation and reasoning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 717–728, Apr. 2018.

[47] J. F. Ruiz, M. Arjona, A. Maña, and C. Rudolph, "Security knowledge representation artifacts for creating secure IT systems," *Comput. Secur.*, vol. 64, pp. 69–91, Jan. 2017.

[48] H. D. Nguyen, N. V. Do, N. P. Tran, X. H. Pham, and V. T. Pham, "Some criteria of the knowledge representation method for an intelligent problem solver in STEM education," in *Applied Computational Intelligence and Soft Computing*. Hindawi, 2020. [Online]. Available: https://www.hindawi.com/journals/acisc/?utm_source=google&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_ADWO_PAI_DYNA_JOUR_X&gclid=EAIaIQobChMIxva40deC6QIV130rCh3jQQ-vcEAAYASAAEgIrJvD_BwE

[49] M. Krótkiewicz, "A novel inheritance mechanism for modeling knowledge representation systems," *Comput. Sci. Inf. Syst.*, vol. 15, no. 1, pp. 51–78, 2018.

[50] J. A. Morente-Molinera, G. Kou, C. Pang, F. J. Cabrerizo, and E. Herrera-Viedma, "An automatic procedure to create fuzzy ontologies from users' opinions using sentiment analysis procedures and multi-granular fuzzy linguistic modelling methods," *Inf. Sci.*, vol. 476, pp. 222–238, Feb. 2019.

[51] X. Guo, S. Wang, D. You, Z. Li, and X. Jiang, "A siphon-based deadlock prevention strategy for S3PR," *IEEE Access*, vol. 7, pp. 86863–86873, 2019.

[52] W. Duo, X. Jiang, O. Karoui, X. Guo, D. You, S. Wang, and Y. Ruan, "A deadlock prevention policy for a class of multithreaded software," *IEEE Access*, vol. 8, pp. 16676–16688, 2020.

[53] S. Aditya, Y. Yang, and C. Baral, "Integrating knowledge and reasoning in image understanding," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Macao, China, Aug. 2019, pp. 6252–6259.

[54] P. Tanwar, T. V. Prasad, and K. Dutta, "Hybrid technique for effective knowledge representation in normal life," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 891–898, 2019.

**HIEN D. NGUYEN** received the B.S. degree in mathematics and informatics and the M.S. degree in computer science from the University of Sciences, VNU-HCM, Vietnam, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from the University of Information Technology, VNU-HCM, in 2020.

He is currently a Lecturer with the Faculty of Computer Science, University of Information Technology, VNU-HCM. His research interests include knowledge representation, automated reasoning, knowledge engineering, especially the intelligent systems in education, such as intelligent problem solver.

Dr. Nguyen received the Incentive prizes of Vietnam Fund for Scientific and Technological Creations (VIFOTEC) Award, in 2016, and the Technological Creations Awards of Binh Duong Province, in 2015.

**NHON V. DO** received the M.Sc. and Ph.D. degrees in computer science from the University of Sciences, VNU-HCM, Vietnam, in 1995 and 2002, respectively.

He was an Associate Professor with the University of Information Technology, VNU-HCM, from 2006 to 2018, and with Hoa Sen University, Vietnam, from 2018 to 2019. He was also an Associate Professor and the Head of the Artificial Intelligence Department, Open University, Ho Chi Minh, Vietnam, from 2019 to March 2020. He has been an Associate Professor and the Head of the Program of Artificial Intelligence, Hong Bang International University, Vietnam, since March 2020. His research interests include artificial intelligence, computer science, and their practical applications, especially intelligent systems and knowledge-based systems.

**VUONG T. PHAM** received the B.S. degree in mathematics and informatics and the M.S. degree in information technology from the University of Sciences, VNU-HCM, Vietnam, in 2003 and 2008, respectively.

From 2003 to 2006, he was a Lecturer with the University of Sciences, VNU-HCM. From 2006 to 2019, he was a Lecturer with the University of Information Technology, VNU-HCM. He also was the Vice Dean of Software Engineering Faculty, University of Information Technology, VNU-HCM, from 2012 to 2019. He is currently a Lecturer at Information Technology Faculty, Sai Gon University, Vietnam. His research interests include artificial intelligence, software engineering, and game development.

**ALI SELAMAT** (Member, IEEE) is currently a Professor in software engineering and the Dean of the Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia. He is also currently the Chair of the IEEE Computer Society Malaysia Section. He has been a Visiting Professor with Hradec-Kralove University, Czech Republic, Kuwait University, and few other universities in Japan. His research interests include cloud-based software engineering, software agents, information retrievals, pattern recognition, genetic algorithms, neural networks and soft-computing, knowledge management, and key performance indicators.

Dr. Selamat is also the Editorial Boards of *Knowledge-Based Systems* (Elsevier), the *International Journal of Information and Database Systems* (IJJIDS) (Inderscience), and the *Vietnam Journal of Computer Science* (Springer).

**ENRIQUE HERRERA-VIEDMA** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1993 and 1996, respectively.

He is a Professor of computer science and the Vice-President for research and knowledge transfer with the University of Granada, Granada, Spain. His H-index is 71 with more than 21 000 citations received in Web of Science and 91 in Google Scholar with more than 31 000 cites received. He has been identified as one of the world's most influential researchers by the Shanghai Center and Thomson Reuters/Clarivate Analytics in both Computer Science and Engineering in the years from 2014 to 2019. His current research interests include group decision making, consensus models, linguistic modeling, aggregation of information, information retrieval, bibliometric, digital libraries, web quality evaluation, recommender systems, and social media.

Dr. Herrera-Viedma is the Vice President for Publications in IEEE SMC Society and an Associate Editor in several journals, such as the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORT SYSTEM, *Information Sciences*, *Applied Soft Computing*, *Soft Computing*, *Fuzzy Optimization and Decision Making*, and *Knowledge-Based Systems*.

• • •