Design an Intelligent Problem Solver in Geometry based on Knowledge Model of Relations

Minh N. Phan, Hien D. Nguyen ✉, Trong T. Le, Dung A. Tran, Nha P. Tran

***Abstract*—Nowadays the intelligent system is used to support learning. The Intelligent Problem Solver (IPS) in education is a system which can solve problems automatically, and its proofs are step-by-step and suitable with the knowledge level of learners. In the course of plane geometry, besides the solution, the figure helps to imagine and grasp the important information from the problem. In this paper, an IPS in Plane Geometry is constructed. This system can give the solution to a problem with its visual figure. The knowledge model of relations (Rela-model) is applied to represent the knowledge base of plane geometry. The relational network representing the relations between geometric objects is the combination between Rela-model and conceptual graph. This network is used to design algorithms for automatic drawing of the figure. Rela-model with related problems is used to design algorithms for solving geometry problems. The IPS in plane geometry can solve common exercises in the mathematical curriculum at Vietnamese middle-school and it is useful to support the learning of pupils.**

***Index Terms*— knowledge representation, intelligent problem solver, knowledge of relations, relational network, automated reasoning, knowledge engineering.**

I. Introduction

NOWADAYS Artificial Intelligence applications can be used in the designing of intelligent systems supporting the learning. An Intelligent Problem Solver (IPS) in education is a part of Intelligent Tutoring System (ITS). This system can solve the problems automatically. Learners only declare the hypothesis and goal of problems based on a sufficient specification language, and the system solves it automatically with the solution is natural and its reasoning is suitable for the knowledge level of the learner [1]. An IPS in education needs to satisfy the requirements as follows [2, 3]:

- The system can solve basic kinds of exercises in the course.
- The specification of input problems is similar to the human. The solutions of the system are readable, and step-by-step.
- The reasoning of this system uses the knowledge of the course. Its solutions are similar to the solving method of the learner.

Plane Geometry is an important mathematical course in middle-school. The solutions of geometry problems require the figure and their reasoning uses the knowledge of the course. The figure helps the learner can observe the problem visually. When solving a problem, the figure gives an image and grasps the important information from the problem. Besides the solution is readable, its reasoning needs to use the geometric knowledge in the course.

In this paper, an IPS in Plane Geometry is designed. This system can give the solution for a problem with its visualizing figure. The knowledge base of geometry is represented by the Rela-model, which is a knowledge model of relations [4, 5]. Based on the structure of this model and the semantic network [6], the relational network between geometric objects in a problem were proposed. This relational network is the foundation to design the automatic drawing the figure of a problem.

Besides, when dealing with a practical problem, firstly, we consider whether we have met a similar or related problem before or not [7]. If so, the solution for the problem can be obtained effectively. Or we determine whether the result of relating problems can be used to solve the practical problem or not [8]. In this paper, Rela-model, which represents the knowledge domain about plane geometry, is also added more to the components of related problems. The improved model, called *Rela-Problem* model, is applied to design the program for solving geometric problems. The algorithms for solving problems uses related problems in the processing, so the reasoning of the program is more effectively and exactly.

The next section analyses some related works for designing of IPS in plane geometry at middle-school. Section 3 presents the knowledge base of plane geometry using the knowledge model of relations. Section 4 design two programs of an IPS in Plane geometry: the program for automatic drawing the figure of a geometry problem, and the

Manuscript received March 29, 2020; revised July 27, 2020. This research is funded by University of Information Technology-Vietnam National University HoChiMinh City, under grant number D1-2020-03.

Minh N. Phan is a lecturer at Software Engineering Faculty, University of Information Technology, Vietnam National University, Ho Chi Minh city (VNU-HCM), Vietnam. Email: minhpn@uit.edu.vn.

Hien D. Nguyen is a lecturer at Computer Science Faculty, University of Information Technology, Vietnam National University, Ho Chi Minh city (VNU-HCM), Vietnam. Email: hiennd@uit.edu.vn. (✉ corresponding author)

Trong T. Le is a Vice Dean of Software Engineering Faculty, University of Information Technology, Vietnam National University, Ho Chi Minh city (VNU-HCM), Vietnam. Email: tronglt@uit.edu.vn.

Dung A. Tran is a lecturer at Software Engineering Faculty, University of Information Technology, Vietnam National University, Ho Chi Minh city (VNU-HCM), Vietnam. Email: dungta@uit.edu.vn.

Nha P. Tran is the Head of Department of Information Technology, University of Transport and Communications, Campus in Ho Chi Minh City, Vietnam. Email: tpnha@utc2.edu.vn.

program for automatic solving the problem. Those programs are built based on the relational network between geometric objects. This network is the foundation to design algorithms for automatic drawing a figure. This section also proposes the knowledge model of relations using related problems, Rela-Problem model. This model is used to design algorithms for automatic solving a geometric problem. The experimental results of the IPS in plane geometry at middle-school are shown in section 5. The last section concludes the paper and gives some future works.

II. Related work

Nowadays there are many programs that can solve the problems in plane geometry. However, those programs still have the limitations for building the IPS in plane geometry.

There are many studies for automated theorem proving with readable proofs [9]. Those methods are classified into three kinds: Search methods [10], Formal Logic methods [11], Coordinate-Free methods [12]. Those results are not applied to build an intelligent system in learning, so the reasoning of their proofs is not suitable for the learner to study the course of plane geometry.

In [13, 14], authors present the area method for automated generating readable machine proofs for geometric theorems: Butterfly theorem, Simson's theorem, Desargues' theorem and Feuerbach's theorem. That method generates readable proofs based on the reasoning on the computing of area. However, its reasoning is not natural, so it is not suitable for supporting the learning of geometric knowledge in the course.

Ontology COKB (Computational Object Knowledge Base) is an ontology that can be represented in many knowledge domains and built practical applications in the IPS system [1]. Ontology COKB using Sample Problems has been used to design the intelligent system for solving problems in plane geometry [15]. Nonetheless, this program cannot draw the figure to illustrate a problem. It also does not trend to support the learning, so it does not meet the requirements of an IPS in education.

An IPS in plane geometry has been constructed in [4] based on the Rela-model. This program can give readable solutions which are step-by-step, reasoning like a human. However, this program cannot product the figure of a problem and its knowledge base is not completeness.

The research in [16] presents an automated system to solve unaltered SAT geometry questions, called GeoS. This system is constructed by combining text understanding and diagram interpretation. It can give the description to a geometric solver that attempts to determine the correct answer. However, this program only works well on SAT questions, it cannot solve problems which require the depth knowledge of geometry.

GeoGebra is an interactive geometry intended for learning and teaching mathematics [17]. Some basic geometric objects can be drawn by the program, such as Point, Segment, Line, Circle. GeoGebra is also used as an automatic reasoning tool for automatic finding of geometric conjectures and the verification or denial of these conjectures through the Relation command [18, 19]. However, it only draws the figure manually, it cannot produce the figure automatically from the hypothesis of the problem. It also cannot solve the geometric problem with a readable solution.

The Geometer's Sketchpad is an interactive software program for searching Euclidean geometry [20]. It can construct the figure alike the drawing with the compass and straightedge method [21]. Nonetheless, this program only can draw a figure manually.

The program in [22] uses the relational network between geometric objects to draw the figure of the problem automatically. However, the program cannot solve the geometry problem.

The result in [23] used the syntax-semantics model to extract relations between the entities and the geometric attributes pattern. Though that, it can represent the semantic of a problem as natural language. Nonetheless, this method has not yet applied to design an IPS in plane geometry.

The study of this paper designs an IPS in Plane Geometry. This system supports the learning of geometry at middle-school. It can give the proof of a problem. The proof includes the figure visualizing the problems and the step-by-step solution with reasoning similar to the learner. The structure of the knowledge model of this system is improved from Rela-model [5] to represent the knowledge domain about plane geometry. Based on the Rela-model with related problems, algorithms for automatic drawing of the figure and solving geometry problems are designed in the system.

III. Knowledge base of Plane Geometry

A. Knowledge model of relations

1) Structure of model

Definition 3.1 [4]: A knowledge model of relations, called Rela-model, consists of three components:

(C, R, Rules)

In which, **C** is a set of concepts. Each concept is a class of objects which have the structure and behaviors for solving their problems. **R** is a set about relations between concepts in **C**, each relation is a binary relation between concepts. **Rules** is a set of inference rules in the knowledge domain.

C – *set of concepts*

Each concept c in **C** has an instance set, called I_c. A concept of Rela-model is a class of objects; each object has the structure as follows:

(Attrs, Facts, RulObj)

In which:

- ***Attrs*** is a set of attributes:

$$\varnothing \neq \boldsymbol{Attrs} \subset \{x_i, I=1..n \mid x_i \in I_{ci}, ci \in \mathbf{C}\}$$

- ***Facts*** is a set of facts of the concept:

$$\boldsymbol{Facts} \subset \{ f \mid f \text{ is a fact, } var(f) \subseteq \boldsymbol{Attrs}\}$$

- ***RulObj*** is a set of rules of the concept:

$$\boldsymbol{RulObj} \subset \{ u \rightarrow v \mid u,v \text{ are sets of attributes, } var(u) \subseteq \boldsymbol{Attrs}, var(v) \subseteq \boldsymbol{Attrs}, u \cap v = \varnothing\}$$

An object also has basic behaviors for solving problems on it. Objects are equipped abilities to solve problems such as: 1/ Determine the closure of a set of attributes; 2/ Determine the closure of a set of facts; and 3/ Execute deduction and give solutions for problems of the form: determine some attributes from some other attributes. These problems have been studied and solved in [4, 5].

Eg. 3.1: In the knowledge domain about plane geometry, the concept QUADRILATERAL $\in$ **C** consists of (Fig. 1):

Attrs = {_A, _B, _C, _D, _a, _b, _c, _d, _m, _n, S, p, ...}

_A, _B: POINT

_a, _b, _c, _d: SEGMENT

_m, _n: SEGMENT

S, p: Real value

where, S and p are the area and the perimeter of the quadrilateral, resp.

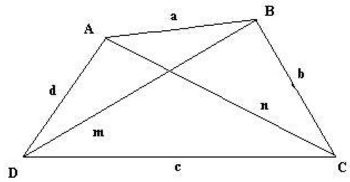

Fig 1. Attributes of a quadrilateral

Facts = {}

RulesObj = {r_1: {m ⊥ n} → {S = ½*m.len*n.len}

r_2: {b // d} → {∠DBC=∠BDA, ∠BCA=∠CAD}}

R – *set of relations*

R is a finite set of binary relations between concepts in **C**. It includes two kinds of relations:

$$\mathbf{R} = R_{hierarchy} \cup R_{relation}$$

- $R_{hierarchy}$ is a set of hierarchical relations on the concepts, and it can be considered as the Hasse diagram for that relation.

- $R_{relation}$ is a set of binary relations between concepts in **C.**

$$\varnothing \neq R_{relation} \subset \{\Phi \mid \Phi \subseteq I_{ci} \times I_{cj}, \mathrm{ci} \in \mathrm{C}, \mathrm{cj} \in \mathrm{C}\}$$

In case $\Phi \in R_{relation}$ is a binary relation on the concept c $\in$ **C**, the properties of Φ are considered: reflexive, symmetric, asymmetric and transitive.

Rules – *set of rules*

Each r $\in$ **Rules** is one of three kinds as follows:

- r is a deductive rule. It has form: u(r) → v(r), where u(r) and v(r) are set of facts of model.
- r is a rule to generate a new object. It is is also a deductive rule: u(r) → v(r), where, u(r) and v(r) satisfy conditions: ∃ object o, o $\in v$(r) and o $\notin u$(r)
- r is an equivalent rule. It has form: h(r), u(r) ↔ v(r), where h(r), u(r) and v(r) are sets of facts, they satisfy conditions: h(r), u(r) → v(r), and h(r), v(r) → u(r) are true.

2) Kinds of facts of Rela-model

Definition 3.2 [5]:

a/ *Classify kinds of facts:*

TABLE I
KINDS OF FACTS

Kind	Function	Specifi-Cation	Condition
1	Inform the object kind	x : c	c $\in$ **C**, x $\in I_c$
2	Determine an object or an attribute of an object	x x.a	x $\in I_c$ (c $\in$ **C**) a $\in$ x.*Attrs*
3	Determine an object by a value or a constant expression	x = <const>	x $\in I_c$ (c $\in$ **C**)
4	Equality on objects	x = y	x $\in I_{cx}$ (cx $\in$ **C**) y $\in I_{cy}$ (cy $\in$ **C**)
5	A relation between objects	x Φ y	Φ $\in$ R

b/ *The unification of two facts:*

Let f1 and f2 be two facts. These facts are unified, ≅, if and only if:

1. f1 and f2 have them same kind k, and
2. if k = 1,2: f1 = f2

   if k = 3: left(f_1) ≅ left(f_2) *and*

   compute(right(f_1)) = compute(right(f_2))

   if k = 4: (left(f_1) ≅ left(f_2) *and* right(f_1) ≅ right(f_1))

   OR (left(f_1) ≅ right(f_2) *and* right(f_1) ≅ left(f_2))

   if k = 5:

   NameofRelation(f_1) ≡ NameofRelation (f_2)

   and

   Property(f_1) ≡ "*symmetric*" *and*

   ((left(f_1) ≅ left(f_2) and right(f_1) ≅ right(f_2))

   or (left(f_1) ≅ right(f_2) and right(f_1) ≅ left(f_2)))

   OR

   NameofRelation(f_1) ≡ NameofRelation (f_2) *and*

   left(f_1) ≅ left(f_2) and right(f_1) ≅ right(f_2);

which:

- compute(expr): return the value of the expression expr.
- NameofRelation(f): return the name of the relation in fact f that is kind 5.
- Property(f): return the property of the relation in fact f that is kind 5.
- left(f): return the left side of expression f.
- right(f): return the right side of expression f.

c/ *Relations on the set of facts:*

Let x be a fact, A and B are sets of facts, the relations between them have been defined as followed:

$$x \odot A \Leftrightarrow \exists g \in A, x \cong g$$

$$A \sqsubseteq B \Leftrightarrow \forall x \in A,\ x \odot B$$

$$A \cong B \Leftrightarrow A \sqsubseteq B \text{ and } B \sqsubseteq A$$

$$A \sqcap B = \{x \mid x \odot A \text{ and } x \odot B\}$$

$$A \sqcup B = \{x \mid x \odot A \text{ or } x \odot B\}$$

$$A \setminus B = \{x \mid x \odot A \text{ and not}(x \odot B)\}$$

3) Model of problem on Rela-model

Definition 3.3: Let $\mathcal{K}$ = **(C, R, Rules)** be a knowledge model of relations as Rela-model. The model of a problem on $\mathcal{K}$ is:

(O, F) → Goal

In which:

O = {O_1, O_2 …, O_m}, the set of objects in the problem.

F = {f_1, f_2,…, f_n}, the set of facts.

Goal = {g_1, g_2,…, g_k}, the goal of the problem.

The algorithm for solving problem on Rela-model has been mentioned in [4, 5]. In the section IV.B, the algorithm using related problem for solving problem is designed. That algorithm is used to find the solution of a geometry problem in the knowledge domain about plane geometry.

B. Knowledge base of Plane Geometry

Based on the knowledge of plane geometry in Vietnamese middle-school, which has been mentioned in the textbooks [25], this knowledge domain can be represented by the Rela-model:

(C, R, Rules)

1) **C**–*set of concepts*

C = {POINT, LINE, SEGMENT, RAY, ANGLE, TRIANGLE and types of it, QUADRILATERAL and types of it, CIRCLE, …}

Eg. 3.2: Concepts PARALLELOGRAM ∈ **C** consists of (Fig. 2):

Attrs = {_A, _B, _C, _D, _O, _a, _b, _c, _d, _m, _n, S, p, ...}

_A, _B, _C, _D, _O: POINT

_a, _b, _c, _d, _m, _n: SEGMENT

S, p: ℝ // real value

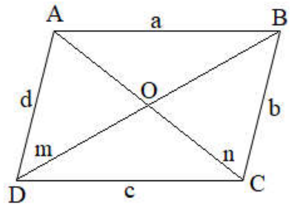

Fig 2. Attributes of a parallelogram

Facts = {a//c, b//d,
_a = _c, _b = _d,
_O = _m intersect _n,
_O_A midpoint _A_C,
_O midpoint _B_D}

RulesObj = {r_1:{ _m ⊥ _n} → {this: ROHMBUS},
r_2:{ _m = _n} → {this: RECTANGLE},
r_3:{ _a = _b} → {this: ROHMBUS},
r_4:{ _m ⊥ _n} → {S = ½*_m*_n}}

2) **R**–*set of relations*

$$\mathbf{R} = \mathbf{R}_{\mathbf{hierarchy}} \cup \mathbf{R}_{\mathbf{relation}}$$

• $R_{hierarchy}$ = {Hierarchical relations on class Triangle, Hierarchical relations on class Quadrilateral}

Fig. 3 represents hierarchy relations on class of Quadrilateral.

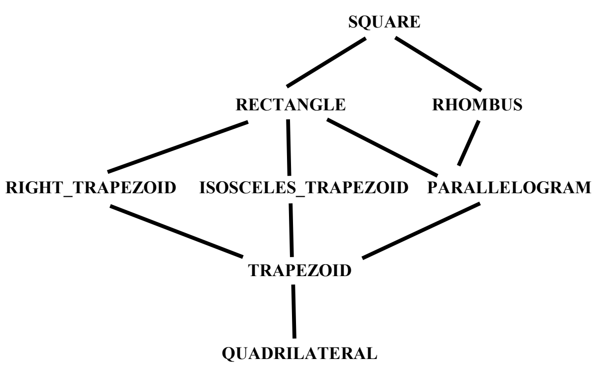

Fig 3. Hierarchy relations on class of quadrilateral

• $R_{relation}$ is a set of binary relations between geometric concepts in **C.**

Some relations between geometric concepts:

- Relations on c ∈ **C**: there are binary relations between objects that have the same type, such as:
  + Relation between Congruent triangles.
  + Relation between Similar triangles.
- Relations on $c_i \times c_j$ (c_i, c_j ∈ **C**): there are binary relations between objects that have different types:
  + Relations on BELONG: a point belongs to a line, a segment, a ray.
  + Relation on MIDPOINT: between a point and a segment.
  + Relations on PERPENDICULAR (⊥): between two lines, two segments, a line and a segment, etc.
  + Relations on PARALLELE (//): between two lines, two segments, a line and a segment, etc.
  + Relations on INTERSECT: between two lines, two segments, a line and a segment, etc.

3) *Rules–set of rules:*

+ Deductive rules:
  - {a: SEGMENT, b: SEGMENT, c: SEGMENT, a ⊥ c, b ⊥ c}
    →{a // b}
  - {O: POINT, AB: SEGMENT, CD: SEGMENT, O = AB intersect CD}
    →{∠AOC = ∠BOD, ∠AOD = ∠BOC}.
  - {ABC: TRIANGLE, M: POINT, N: POINT, M midpoint AB, N midpoint AC}
    →{MN // BC, MN = ½ BC}
+ Rules which generates a new object
  - {a: LINE b: LINE, a intersect b}
    →{N: POINT, N = a intersect b}
+ Equivalent rule:
  - {MNPQ: QUADRILATERAL, O: POINT, O = MP intersect NQ}

{MNPQ: PARALLELOGRAM} ↔ {O midpoint MP, O midpoint NQ}

IV. DESIGN AN INTELLIGENT PROBLEM SOLVER IN PLANE GEOMETRY

In an IPS in Plane Geometry, the solution of problems includes the figure and reasoning steps for solving them. The figure visualizes the relations between geometric objects of the problem. It helps the learner to understand and follow the reasoning. The reasoning steps are natural, and use the knowledge of the corresponding course. The solutions are similar to the reasoning of human when solving the current problem.

A. Design the program for automatic drawing the figure of problems

In plane geometry, the figure illustrates the relations between geometric objects in a problem. Though the knowledge base of plane geometry in section 3.B, those relations can be represented by the relational network. This network is built based on the combination between the structure of Rela-model and the conceptual graph [22, 5].

1) *Relational Network of Geometric objects*

Definition 4.1: Given a knowledge base of plane geometry $\mathcal{K}$ = **(C, R, Rules)** as Rela-model, and a problem P = **(O, F) → Goal** as Def. 3.3. The relational network represents the relations between geometric objects in the problem P is a graph:

(Vertex, Arc)

In which:

• **Vertex** is a set of vertices of the graph. Each vertex is a geometric object. There are two kinds of objects:

* *Primitive object:* the object can be determined freedom or based on other objects.

* *Deductive object*: the object is implicitly determined if and only if its attributes are determined.

Eg. 4.1:

- POINT and LINE are concepts which are classes of primitive kind objects.

- ANGLE is a concept of the deductive kind because objects of it are determined if and only if their two rays are defined.

- TRIANGLE is a concept of the deductive kind because objects of it are determined if and only if their three vertices, which are points, are defined.

- **Arc** is a set of arcs which represent relationships between objects. Those arcs are used to connect the vertices of the relational network. There are two kinds of arcs:

  * *Connection arc* (⟶): Represents the connections between primitive objects.

  * *Implicitly arc* (┈▷): represents the connection with the deductive object.

Eg. 4.2: Let ABCD be a quadrilateral. The relational network between geometric objects of quadrilateral ABCD is as Fig. 4:

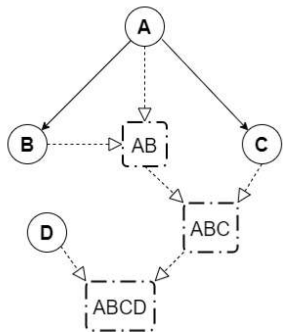

Fig 4. Relational network of quadrilateral ABCD.

2) Alogrithms for automatic drawing the figure of plane geometry

Given a knowledge base of plane geometry $\mathcal{K}$ = **(C, R, Rules)** as Rela-model, and the problem P = **(O, F) → Goal** as Def. 3.3. There are two problems for drawing the figure of the problem P.

- Problem 1: Build the relational network G representing the relations between geometric objects in the problem P.
- Problem 2: Given a relational network G, export the the figure of gemometric objects and their relations in the G.

Algorithm 4.1: *Algorithm for solving Problem 1*

Input: Knowledge base of plane geometry K = **(C, R, Rules)** as Rela-model.
Problem P = **(O, F) → Goal**
Output: Relational network G of geometric objects in the problem P.

G is an empty graph
For each fact *f* in **F** do
1. Create sub network based on fact *f*.
For each object *o* in *f* **do**
{
If object *o* does not depend on other objects **then**
Object o is an isolated vertex.
Else if object o is a deductive object **then**
Add the implicitly arcs from dependent objects to object *o*.
Else
Add the connection arcs from dependent objects to object *o*.
}
2. Merge sub-network into G.
od;
Return network G;

Algorithm 4.2: *Algorithm for solving problem 2*

Input: Knowledge base of plane geometry K = **(C, R, Rules)** as Rela-model.
Problem P = **(O, F) → Goal.**
Relational network G represents objects in the problem P, which was established by Algo. 4.1.
Output: A figure of the problem P.

For each vertex *v* in **G.Vertex do**
If there are not any arc to *v* **then**
v is generated freely.
Elif *v* depends on other vertex **then**
v is drawn based on other objects.
Elif *v* is an object of deductive kind **then**
v is determined by previous objects.
(All objects are determined by defining their coordinates or equations).
od;
Show the figure

B. Design the program for automatic solving the problem in Plane Geometry

In order to deal with a practical problem, we often search related problems which have been solved before, and then proposing an appropriate solution for the problem. Besides, the result of related problems can be used for solving the current problem. For enhancing the effectiveness of the reasoning to solve the problem, the reasoning process uses the knowledge of related problems as the experiment of human when solving the current problem.

1) Related Problem

The *related problem* is the problem related to the current problem. It has the short and simple solution, but it has the high frequency for using in the knowledge domain [15, 24].

Definition 4.2: Given the knowledge domain $\mathcal{K}$ = **(C, R, Rules)** as Rela-model.

a/ A related problem is a solved problem on the knowledge $\mathcal{K}$, it consists of three components as follows:

$$(\mathbf{O_P}, \mathbf{F_P}, \mathbf{D_P})$$

In which: (O_P, F_P) is hypothesis of the related problem.
$\mathbf{D_P}$ is a list of rules that can be applied on (O_P, F_P).

b/ *Criteria of a related problem:* Two criteria are determining if a problem P is a related problem [8, 15]:

- Frequency of using problem P in the knowledge domain is high: In testing problems of the knowledge domain, frequency of using problem P is greater than some δ.
- Number of Objects in problem P is small: $card(O_P) \le \lambda$.

In which, (δ, λ) are constants depending on the knowledge domain. In the knowledge domain about plane geometry, δ= 0.5 and λ=5.

Eg. 4.2: *Problem about solving the right triangle.*

"Given a right triangle DEF, it has D = 90^o, and DH is an altitude of triangle DEF.

Let DE = c, EF = a. Compute the length of DH."

_The specification of the related problem:

O_P := {DEF: RightTriangle, DH: Segment}

F_P "= {∠EDF = 90, DE = c, EF = a,

DH altitude RightTriangle(ABC)}

D_P := [s_1, s_2, s_3, s_4].

s_1 = {DEF: RightTriangle, ∠EDF = 90} → {$EF^2 = DE^2 + DF^2$}

s_2 = {DEF: RightTriangle, DH altitude RightTriangle(DEF)} → {DH.EF = DE.DF}

s_3 = {DE = c, EF = a, $EF^2 = DE^2 + DF^2$} → { $DF = \sqrt{a^2 - c^2}$ }

s_4 = {DE = c, EF = a, $DF = \sqrt{a^2 - c^2}$, DH.EF = DE.DF}

$$\rightarrow \left\{ DH = \frac{c\sqrt{a^2 - c^2}}{a} \right\}$$

Definition 4.3: Knowledge model of relations with Related Problems, called *Rela-Problem*, consists of components as follows:

(C, R, Rules) + Problems

In which:

- **(C, R, Rules)** is the knowledge model as Rela-model.
- **Problems** are a set of Related Problems of this knowledge domain.

2) Algorithms for automatic solving problem in plane geometry

Algorithm 4.3: *Finding a related problem.*

Given the knowledge domain about plane geometry $\mathcal{K}$ = **(C, R, Rules) + Problems** as Rela-Problem model.

Input: The problem P = (O, F) → Goal on this knowledge domain as Def. 3.3.

Output: The related problem can be applied on P.

Step 1. Related := **Problems;**
Related_Found := false;

Step 2. **Repeat**
Select S in Related.
if (S.O_P ⊑ P.O and S.F_P ⊑ P.F) then
Related_Found := true;
end.
Related := Related \ S
Until Related = {} or Related_Found

Step 3. **If** Related_Found **then**
S is a related problem of the problem;
else
There is no any related problem found;

Algorithm 4.4: *Finding the solution of the problem.*

Given the knowledge domain about plane geometry $\mathcal{K}$ = **(C, R, Rules) + Problems** as Rela-Problem model.

Input: The problem P = (O, F) → Goal on this knowledge domain.

Output: The solution of the current problem P.

The reasoning of this algorithm uses the forward chaining to solve the current problem. It also combines heuristics and related problems in the reasoning process. Objects attend the reasoning as active agents for solving some sub-problems on themselves.. Hence, the system can run more quickly and effectively.

Step 0: Initialize variables
flag := true;
KnownFacts := **F**;
count :=0; # the number of new objects are generated.
Sol :=[ ]; # solution of problem

Step 1. Collect the elements in hypothesis and goal part of problem.
Classify kind of facts in *KnownFacts*

Step 2. Find the **related problem S** can be applied by using algorithm 4.3.
Update S.D_P into *Sol* and *KnownFacts*.
If Goal is obtained **then**
Go to step 5.

Step 3: Using behaviours of each object in set **O** and set of facts **F** to determine new facts.
If Goal is obtained **then**
Go to step 5.

Step 4. Use heuristic rules to select a rule in set **Rules** for getting new facts or new objects.
while (*flag* != false) and **not(Goal** is determined) **do**
Search *r* in **Rules**-set which can be applied to *KnownFacts*.
4.1. Case: *r is a deductive rule*
Update *KnownFacts* and *Sol*;
continue;
4.2. Case: *r is a rule for generating a new object*
if *count* <= card(**O**) **then** #only generate at most number of objects in hypothesis
if (*r* generates a new object *o*) and
not(*o* ⊙ *KnownFacts*) **then**
count:=*count*+1;
Update *KnownFacts* and *Sol*;
Goto Step 3 with new object *o*;
end if;
end if; #4.2
4.3. Case: *r is an equivalent rule*
Update *KnownFacts* and *Sol*;
continue;
4.4. if (rule *r* cannot be found) **then**
flag := false;
end if;
end do; #while

Step 5. Conclusion of problem
if G is determined **then**
Problem **(O, F) → G** is solvable;
Sol is a solution of problem;
else
Problem **(O, F) → G** is unsolvable;
end if;

V. TESTING AND EXPERIMENTAL RESULTS

A. Testing

The knowledge domain about plane geometry in middle-school has been mentioned in textbooks [25]. This knowledge base is organized in section III.B. Based on this knowledge base, the main functions of an intelligent problem solver in plane geometry are designed in section IV: The automatic drawing of the figure and the reasoning to solve the geometry problems. This program can solve some basic and advanced exercises in plane geometry. The solution includes the figure visualizing the hypothesis of the problems, and the reasoning steps being natural, similar to the reasoning of human, suitable the knowledge level of the user. This program is useful for the studying of the students in plane geometry at the middle school.

Eg. 5.1: Let ABCD be a trapezoid, F is the midpoint of BC, K is the midpoint of BD, and E is the midpoint of AD.

Prove: F belongs to the line (EK).

+ Model of problem: **(O, F) → Goal**, where:

O = {ABCD: Trapezoid, F: Point, K: Point, E: Point}

F = {F midpoint BC, K midpoint BD, E midpoint AD}

Goal = {"Prove": F belong line(EK)}

+ The proof of the system:

The figure of the problem:

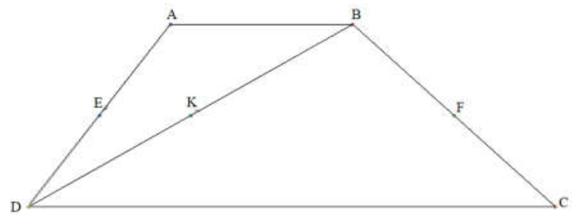

Fig 5. The figure of the program for Eg. 5.1

The solution of the problem:

Step 1: {ABCD: Trapezoid} → {AB // CD}
Step 2: {DAB: Triangle, E: Point, K: Point,
E midpoint AD, K midpoint BD}
→{EK // AB}
Step 3: {BDC: Triangle, K: Point, F: Point,
K midpoint BD, F midpoint BC}
→{KF // CD}
Step 4: {EK // AB, AB // CD}→ {EK // CD}
Step 5: {EK // CD, KF // CD} → {F belongs line(EK)}

Eg. 5.2: Let ABCD be a parallelogram, and point O is the intersection point of AC and BD. Point M belongs to AB. Point N is the intersection point of line(OM) and CD.

Prove: OM = ON.

+ Model of problem: **(O, F) → Goal**, where:

O = {ABCD: Parallelogram, O: Point, M: Point, N: Point}

F = {O = AC intersect BD, M belongs AB,
N = line(OM) intersect CD}

Goal = {"Prove": OM = ON}

+ The proof of the system:

The figure of the problem:

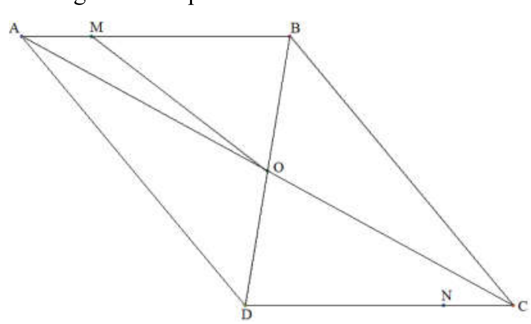

Fig 6. The figure of the program for Eg. 5.2

The solution of the problem:

Step 1: {ABCD: Parallelogram, O: Point
O = AC intersect BD}
→ {O midpoint AC, ∠BAC = ∠ACD}
Step 2: {O = AC intersect BD} → {O belongs AC}
Step 3: {N = line(OM) intersect CD}
→ {N belongs line(OM), N belongs CD}
Step 4: {N belongs line(OM)} → {O belongs MN}
Step 5: {O belongs AC, O belongs MN}
→ {O = AC intersect MN}
Step 6: {O midpoint AC} → {OA = OC}
Step 7: {N: Point, AC: SEGMENT, MN: SEGMENT,
O = AC intersect MN}
→{∠AOM = ∠CON}
Step 8: {O midpoint AC} → {OA = OC}
Step 9: {OAM: Triangle, OCN: Triangle,
OA = OC, ∠AOM = ∠CON, ∠BAC = ∠ACD}
→ {Triangle(OAM) = Triangle(OCN)}
Step 10: {OAM: Triangle, OCN: Triangle,
Triangle(OAM) = Triangle(OCN)}
→ {OM = ON}

C. *Experimental results*

The exercises for testing is collected from workbooks in [25]. Those problems are classified by grades from 6th – 9th:

- 6th grade: The problems are about geometric objects: line, segment, ray, angle and their relations.
- 7th grade: Besides geometric objects in the 6th – grade, the problems are about: triangle and their properties: altitude, bisector, etc.
- 8th grade: Besides the knowledge of triangle, the problems are about: quadrilateral and their properties, kinds of quadrilateral: parallelogram, rhombus, rectangle, square, etc.
- 9th grade: Besides the knowledge of triangle and quadrilateral, the problems are about: circle and their properties, kinds of circle: circumscribed circle, inscribed circle, escribed circle.

There are many advanced kinds of problems in 8th and 9th grades. In this testing, we only test on basic kinds of problems in those grades.

TABLE II
RESULTS OF TESTING

Grade	Number of testing problems	Number of correct problems	Proportion
6th grade	19	13	68%
7th grade	17	10	59%
8th grade	16	10	62%
9th grade	25	20	80%
Total	**77**	**53**	**69%**

In 6th and 7th grades, the problems only have single inquires for each geometric object, so they are not enough data for drawing them exactly. In 8th and 9th grades, our program can solve almost basic problems. However, some problems have the complex relations between objects, so the program have not yet solved them.

D. *Comparison with another program*

GeoGebra is free dynamic geometry software [17, 19]. It includes automated reasoning features based on computer algebra algorithms. It is a relational tool allows the user to numerically compare various geometric objects, plane geometry objects like lines and segments can be compared symbolically as well, and rigorous facts on equality, parallelism, perpendicularity, collinearity or concyclicity can be obtained in a user-friendly way.

The program in [26] is a program for proving plane geometry theorems stated by text and diagram in a complementary way. This program uses a syntax-semantics model to extract the geometric relations from theorem text. Then, the program of theorem proving is built by using the existing proving methods which take the extracted geometric relations as input.

Table III compares GeoGebra, the program in [26] and our program based on the criteria of an IPS in the learning of plane geometry: the method for input an exercise, the solution of program, the ability to support the learning.

E. Results of the survey about the ability to support the learning

Our program has been also examined by 86 students at 9th-grade of two middle-schools in Ho Chi Minh city, Vietnam. This survey is interested in four criteria: User-friendly Interface, Sufficient knowledge base, The ability to solve problems, Usefulness.

Firstly, each student selects 02 exercises from a set of solvable exercises (53 exercises). He/She checks their solutions. Secondly, the student inputs to another problem. The program solves and shows solutions for that exercise. Finally, they assess each criterion with a level from 1 – 5, respectively from very bad – very good. The meaning of each level is presented as Table IV [29, 30].

TABLE IV
THE MEANING OF EACH LEVEL OF CRITERIA

Level	Meaning
1	**Absolutely inadequate** The program does not satisfy the requirements of the criterion.
2	**Inadequate, improvements necessarily.** The program tends to the requirement of the criterion; however, it needs to be major improved.
3	**Inadequate, some minor improvements need to be revised.** The requirement of the criterion is almost adequate, but some minor improvements need to be revised to make it adequately.
4	**Adequate as expected.** The program satisfies the requirements of the criterion fully.
5	**Better than adequate.** The program works better than expected.

The result of this survey is shown in Table V.

TABLE V
RESULTS OF SURVEY

Criteria	**Level** (Very bad → Very good)				
	1	2	3	4	5
User-friendly interface		28%		72%	
Sufficient Knowledge		24%		76%	
Ability to solve problem		29%		71%	
Usefulness		19%		81%	

Our program got the good feedback from some students for supporting their learning. It is useful to support the learning of students in plane geometry. However, some hard exercises have not yet been solved by our program, because those problems require the depth knowledge to solve them. Our program is emerging to develop a practical intelligent software to release in the real-word.

VI. CONCLUSION AND FUTURE WORK

In this paper, an Intelligent Problem Solver in plane geometry is designed. The system can give the solution for a problem with its visualizing figure. The knowledge domain about plane geometry is represented by the knowledge model of relations, called Rela-model.

The figure is an essential factor in the geometry problems. For designing the automatic drawing of the figure, a relational network between geometric objects is given by the combining between Rela-model and conceptual graph. The network, algorithms for automatically drawing the figure of the problem are proposed. The program can draw almost basic geometric objects in middle-school, such as point, line, triangle, Quadrilateral, circle. The figure generating by this program is precise and can be understood by learners. Besides, for designing the automatic solving the geometry problem, Rela-model has been extended by adding the component of related problems. The extension model is called *Rela-Problems* model. Using Rela-Problems model, the reasoning for solving the geometry problem is more effective. The solutions of the program are also step-by-step natural, precise and simulate the human reasoning. The IPS in plane geometry is useful for the studying of the students in plane geometry at the middle school.

In the future, the method for extracting the information of geometric problems from natural language will be studied. This method is researched based on the grammar structure of language [31]. The current system combining the processing of natural language will be developed to a practical intelligent software for supporting the learning of middle-shool students in mathematics in the real world. This program is also studied more for solving some advanced kinds of problems, especially the problems in the exam of high-school enrollment.

Moreover, the relational network will be studied to represent the relations between objects in solid geometry. From that, a program for drawing the figure in solid geometry will be designed. The combination of this program and the IPS in solid geometry [32] makes a completely IPS in solid geometry. It can give the solution and the figure of the geometry problem, and that IPS can meet the requirements of an IPS in education [2, 3].

REFERENCES

[1] N.V. Do, “Chapter 6: Intelligent Problem Solvers in Education: Design Method and Applications”, in *Intelligent Systems*, V.M. Koleshko (Eds.), Intech, 2012, pp. 121-148.

[2] I. Hatzilygeroudis, J. Prentzas, “Chapter 9: Knowledge Representation in Intelligent Educational Systems”, in *Web-Based Intelligent E-Learning Systems: Technologies and Applications*, Zongmin Ma (Eds.), IGI publisher, 2006.

[3] H.D. Nguyen, N.V. Do, P.N. Tran, X.H. Pham, V.T. Pham, “Some criteria of the Knowledge Representation method for an Intelligent

Problem Solver in STEM education", *Applied Computational Intelligence and Soft Computing*, Vol. 2020, Article ID 9834218, 2020.
[4] H.D. Nguyen, V.T. Pham, T.T. Le, D.H. Tran, "A Mathematical Approach for Representation Knowledge about Relations and Its Application', in *Proc. 2015 IEEE International Conference on Knowledge and Systems Engineering*, Ho Chi Minh, Vietnam, 2015, pp. 324-327.
[5] N.V. Do, H.D. Nguyen, A. Selamat, "Knowledge-Based model of Expert Systems using Rela-model", *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, Vol. 28, No. 8, 2018, pp. 1047 – 1090.
[6] J.F. Sowa, A. Borgida, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, 1991.
[7] G. Polya, *How to solve it*, Publisher of Education, 1997.
[8] N.V. Do, H.D. Nguyen, T.T. Mai, "Designing an Intelligent Problems Solving System based on Knowledge about Sample Problems", in *Proc. 5th Asian conference on Intelligent Information and Database Systems (ACIIDS 2013)*, Kuala Lumpur, Malaysia, 2013, LNAI 7802, Springer, pp. 465-475.
[9] J. Jiang, J. Zhang, "A Review and Prospect of Readable Machine Proofs for Geometry Theorems", *Journal of System Science and Complexity*, 25, 2012, pp. 802-820.
[10] S.C. Chou, X.S. Gao, J.Z. Zhang, "A deductive database approach to automated geometry theorem proving and discovering", *Journal of Automated Reasoning*, Vol. 25, No. 3, 2000, pp. 219–246.
[11] C. Dehlinger, J. Dufourd, P. Schreck, "Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry", in *Automated Deduction in Geometry,* Richter-Gebert, J., Wang, D.M. (Eds.), LNAI 2061, Springer-Verlag, Berlin, Heidelberg, 2001.
[12] Y. Zou, J.Z. Zhang, "Automated generation of readable proofs for constructive geometry statements with the mass point method", in *Automated Deduction in Geometry,* P. Schreck, J. Narboux, J. Richter-Gebert (Eds.), LNAI 6877, Springer-Verlag, Berlin, Heidelberg, 2011.
[13] S.C. Chou, X.S. Gao, J.Z. Zhang, *Automated Production of Traditional Proofs for Theorems in Euclidean Geometry, IV, A Collection of 400 Geometry Theorems*, TR-92-7, Department of Computer Science, WSU, 1992.
[14] P. Janicic, J. Narboux, P. Quaresma, "The area method: A recapitulation", *Journal of Automated Reasoning*, Vol. 48, No. 4, 2012, pp. 489–532.
[15] N. Do, H. Nguyen, "A Reasoning method on Knowledge Base of Computational Objects and Designing a System for automatically solving plane geometry problems", Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2011, WCECS 2011, 19-21 October, 2011, San Francisco, USA, pp. 294-299.
[16] M. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, C. Malcolm, "Solving Geometry Problems: Combining Text and Diagram Interpretation", in *Proc. 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, 2015.
[17] GeoGebra (2020, March 21). Available: https://www.geogebra.org/geometry
[18] M. Hohenwarter, Z. Kovács, T. Recio, "Using Automated Reasoning Tools to Explore Geometric Statements and Conjectures", in *Proof Technology in Mathematics Research and Teaching*, G. Hanna, D.A. Reid, M. Villiers (Eds.), Springer, 2019, pp.215-236.
[19] F. Botana, Z. Kovács, A. Martínez-Sevilla, T. Recio, "Automatically Augmented Reality with GeoGebra", in *Augmented Reality in Educational Settings*, T. Prodromou (Eds.), Brill, 2019, pp. 347-368.
[20] C. Kimberling, *Geometry in Action: A Discovery Approach Using The Geometer's Sketchpad*, Key College Publishing, Emeryville, California, USA, 2003.
[21] Geometer Sketchpad (2020, March 21). Available: https://www.dynamicgeometry.com/
[22] H.D. Nguyen, K. Le, T.P. Luc, N.M. Phan, V.T. Pham, "Model for representation the relations between geometric objects and Drawing a figure of the problem in Plane geometry", *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol. 8, No. 6, 2019, pp. 3081-3086.
[23] W. Gan, X. Yu, "Automatic understanding and Formalization of Natural Language Geometry problems using syntax-semantics models", *International Journal of Innovative Computing, Information and Control*, Vol. 14, No. 1, 2018, pp. 83-98.
[24] N. Do, H. Nguyen, "A reasoning method on Computational Network and Its applications", Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2011, IMECS 2011, 16-18 March, 2011, Hong Kong, pp. 137-141.
[25] Vietnam Ministry of Education and Training (2015), *Textbook and workbook of middle school Geometry*, Publisher of Education.
[26] G. Wenbin, Y. Xinguo, Z. Ting, W. Mingshu, "Automatically Proving Plane Geometry Theorems Stated by Text and Diagram", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 33, No. 7, 2018.
[27] H. Nguyen, N. Do, V. Pham, A. Selamat, E. Herrera-Viedma, "A method for knowledge representation to design Intelligent Problems Solver in mathematics based on Rela-Ops model", *IEEE Access*, Vol. 8, pp. 76991–77012, 2020.
[28] G. Wenbin, Y.Xinguo, W.Mingshu, "Automatic Understanding and Formalization of Plane Geometry Proving Problems in Natural Language: A Supervised Approach", *International Journal on Artificial Intelligence Tools*, Vol. 28, No. 4, 2019.
[29] H. Nguyen, N. Do, "Intelligent Problems Solver in Education for Discrete Mathematics", in *Proc. of 16th International Conference on Intelligent Software Methodologies, Tools, and Techniques* (SOMET 2017), Kitakyushu, Japan, Sept. 2017, Frontiers in Artificial Intelligence and Applications, vol. 297, pp. 21-34.
[30] E.A. Giakoumakis, G. Xylomenos, "Evaluation and Selection criteria for software requirements specification standards", *Software Engineer Journal*, 11(5), pp. 307-319, 1996.
[31] H.D. Nguyen, T. Huynh, S.N. Hoang, V.T. Pham, I. Zelinka, "Language-oriented Sentiment Analysis based on the grammar structure and improved Self-attention network", in *Proc. of 15th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE 2020), pp. 339-346, May 2020.
[32] H.D. Nguyen, D. Nguyen, V.T. Pham, "Design an Intelligent Problems Solver about Solid Geometry based on Knowledge model about Relation", in *Proc. 2016 IEEE International Conference on Knowledge and Systems Engineering (KSE 2016)*, Ha Noi, Vietnam, 2016, pp. 150-155.

TABLE III
COMPARISON WITH OTHER PROGRAMS

Criteria	GeoGebra [17]	The program in [26]	Our program
Method of input	Problems are inputted through the figure. However, it only show the basic geometric objects and cannot extract the relations between them by that figure.	Problems are inputted by natural language. The program can extract relations between geometric objects from natural language by using S^2 model [26, 28].	Problem are inputted by the specification language. This language is similarly to the human language.
Solution of program	The knowledge base of the program about geometric objects throught on the algebraic equations. It cannot solve the exercises in plane geometry. It can understand the relations between of geometric objects based on computer algebra.	The reasoning is a combination between forward chaining search method and algebraic method. The proof is not pedagogical. It is not step-by-step as the human reasoning, especially it does not a figure to illustrate the problem.	The knowledge base of the program is organized as the human knowledge. The reasoning also uses heuristic rules for solving problems. The proof is pedagogical. It is readable, step-by-step. It simulates the reasoning of human in the processing. It also has a figure to illustrate the problem.
Supporting the learning	The program supports the learning of students in geometry. It helps students more understanding the geometric objects visually. It also supports Augmented Reality (AR) for some 3D-objects.	The program does not tend to support the learning. Hence, its results are not suitable for the learning of users.	The program is an intelligent system in learning. It tends to support the learning of users. The program meets requirements of an intelligent problem solver in education, especially in studying of mathematics [27].

MODIFICATION

- Date of modification: December 08, 2020
- Update the information of this paper are placed above the author details on the first page:

“Manuscript received March 29, 2020; revised July 27, 2020.”