A Program for Solving Problems in Vector Algebra and Analytic Geometry

Nhon Do, Diem Nguyen, Trang Nguyen

***Abstract*—**nowadays, there are some software programs for Mathematics. However, they aren't easy for students to use and they have no a detail solution. In this paper, we'll introduce an application which can solve those problems. We researched to solve problems in vector algebra and analytic geometry by using some techniques for construction and design depend on KBCO model. The program is implemented in Maple and C#. We have tested and evaluated seriously. As a result, we came to the following conclusion: this application can automatically solve problems and give human readable solutions that similar to those written by teachers and students.

***Keywords*—**artificial intelligence, KBCO model for Vector Algebra, KBCO model for Analytic Geometry, knowledge base.

I. Introduction

MATHEMATICS is a science which is applied for many fields such as business, management, education. So that applied informatics in Math has an important meaning. There are some software programs for Math but they only support for simple computing; giving results with complex guide on Symbolic computation; does not have a friendly user interface. There are also some scientific researches projects in this field and they have some good results: Using Mathematics software program for teaching and scientific researches [9], A system supports for studying knowledge and solving Analytic Geometry problems [4], The Extensive Computational Networks and Applying in Educational software [5]. Some theses applied KBCO model for the knowledge bases [4]-[5] to solving problems, especially model for Solving problems in two-dimensional Analytics Geometry, have some achievements about modeling a problem class, design algorithm, setting processes, finding solution for a problem class base on a knowledge base; researching and development some fact kinds, process some facts relative function knowledge.

However, they only solved for small, simple problems without solving general problem class, problems depend on argument is not implement; user interface is not visual, it's difficult for using; not giving a human readable solutions.

In this paper, we present how to use Knowledge Representation methods to implement a program for solving problems in Vector Algebra and two-dimensional Analytics Geometry, which are applied previous achievements scientifically to surmount those problems. We use two sub models of KBCO model to represent knowledge, to model problems and to design algorithms. This gives us useful models to construct components and the whole system. By this way we implemented the program in Maple, designed user interface in C# environment and it is easy to use.

Nhon Do is with the Department of Computer Science of Information Technology University, Ho Chi Minh City, Vietnam (phone: +84-908107799; e-mail: nhondv@uit.edu.vn).

Diem Nguyen is with the Department of Computer Science of Information Technology University, Ho Chi Minh City, Vietnam (phone: +84-0932091060; e-mail: nguyenthingocdiemmhd@gmail.com).

Trang Nguyen is with the Department of Computer Science of Information Technology University, Ho Chi Minh City, Vietnam (phone: +84-01267841382; e-mail: thutrangitmt@gmail.com).

II. Model for Vector Algebra knowledge

A. Model

Building a model depend on KBCO (Knowledge Base of Computational Objects) [2] consists of 3 components: **(C, R, Rules)**. We call this model "KBCO model for Vector Algebra".

Where:

- **C** is a set of concepts of computational objects (Com-objects [2]). Consist of: "Point", "Vector", "Segment", "Triangle", "Parallelogram".
    - o Point is a basis object.
    - o Segment, Vector are objects of the first level.
    - o Triangle, Parallelogram is object of the second level.
- **R** is a set of relations on the concepts. Each relation is identified by <the name of relation> and the kinds of objects. Relation has one of following properties, such as: reflex, symmetry, bridge.
    - o Example: Relation "Centroid" between a "Point" and a "Triangle": [Centroid, Point, Triangle],{}
- **Rules** is a set of rules. Almost properties, clauses, theorems in Algebra Vector can be represented by rules on facts relating to Com-objects.
    - o Example: {A,B,C,G : Point, G is centroid of triangle ABC]} →{ $\overrightarrow{GA}+\overrightarrow{GB}+\overrightarrow{GC}=\overrightarrow{0}, \overrightarrow{AG}+\overrightarrow{BG}+\overrightarrow{CG}=\overrightarrow{0}$}

In the "KBCO model for Vector Algebra", we use 5 fact kinds:

- **Fact of kind 1**: information about object kind.
    - o Structure: [<object>, <the kind of object>].
    - o Example: [a,"Vector"].
- **Fact of kind 2**: a determination of an object or an attribute of an object.
    - o Structure: <object> or <object>.<property>

- Example: Segment[A,B], Segment[A,B].x.

- **Fact of kind 4**: equality on objects or attributes of objects.
  - Structure: <object>|<object>.<property> = <object>|<object>.<property>.
  - Example: O1.c = Segment[A,B], a = b. E.A1=[-a,0] ('a' can be object or property of object)
- **Fact of kind 5**: a dependence of an object or an attribute of an object on other objects by a formula.
  - Structure: <object> = <expression of objects or expression of properties of objects> or <object>.<property> = <expression of objects or expression of properties of objects>
  - Example: a = 2*c + b
- **Fact of kind 6**: a relation on objects or attributes of objects.
  - Structure: [<the name of object>,<object1>,<object2>,…]
  - Example: ["Midpoint",M,Segment[A,B]]

B. Organization of Knowledge Base

The knowledge base is organized by "KBCO model for Vector Algebra" as the figure below:

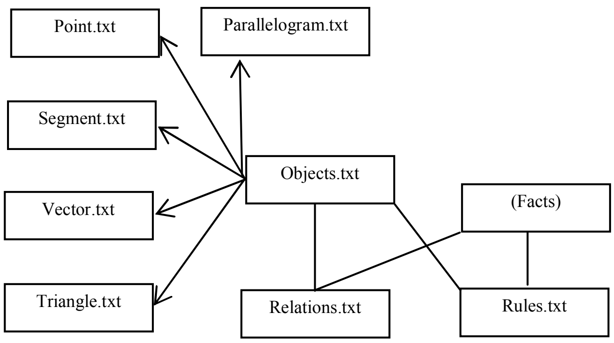

Fig. 1 Organization of Vector Algebra knowledge domain

Where:

- The file "*Objects.txt*": stores names of concepts: Point, Segment, Vector, Triangle, Parallelogram.
- The files <name of concept>.txt store the specifications of structures of Com-objects (there are 5 files are "*Point.txt*", "*Segment.txt*", "*Vector.txt*", "*Triangle.txt*", "*Parallelogram.txt*").
- The file "*Rules.txt*" stores deductive rules.

The file "*Relations.txt*" stores specification of deferent relations in Com-objects net.

C. Model for Problems

From the general form of problems in Vector Algebra consists of hypothesis part and goal part. Hypothesis part has a set O of Com-objects [2], and a set Facts of hypothesis facts. Goals which is target facts of the problem can be some attributes, some objects. We denote the problem by: **(O, Facts, Goals)**

Example: Triangle MNP, Median MQ, M is midpoint of MQ. Proof: 2*Vector[R,M] + Vector[R,N] + Vector[R,P] = Vector[0].

Summary:

- O={M,N,P,Q,O,R: Point, O1: Triangle}
- Facts={[O1,"Triangle[A,B,C]"],["Median",Segment[M,Q], Triangle[M,N,P]],["Median",R,Segment[M,Q]}
  Goals:=[2*Vector[R,M]+Vector[R,N]+Vector[R,P]= Vector[0]]

D. Algorithms

First, begin with recording the information of the problem. Then because there are two problem forms mentioned in Vector Algebra knowledge, we have two following algorithms:

1. Algorithms to Simplify Vector Expression (Algorithms1)

Vector expression is an expression whose elements are vectors and operators are addition, subtraction between vectors. Example: we have following expressions such as: $\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}$,$\overrightarrow{AD}+\overrightarrow{BE}+2*\overrightarrow{CF}$, CF, $\overrightarrow{AD}+\overrightarrow{BE}-\overrightarrow{CF}$.

- **Step 1**: Change all of subtraction in Vector expression into expression with only addition vector by adding property of opposite vector. Ex: $-\overrightarrow{CD}=\overrightarrow{DC}$
- **Step 2**: Apply rules such as: three point rule, vector addition to simplify Vector expression and get a new Vector Expression.
  - If that new Vector Expression is $\vec{0}$ or can't find rule can apply then stop.
  - Else do step 2 again

Example: Give 6 points A, B, C, D, E, F. Simplify: $\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}-\overrightarrow{AE}-\overrightarrow{BF}-\overrightarrow{CD}$

- Solution found by human:
  Do:
  $$\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}-\overrightarrow{AE}-\overrightarrow{BF}-\overrightarrow{CD}=\left(\overrightarrow{AD}-\overrightarrow{AE}\right)+\left(\overrightarrow{BE}-\overrightarrow{BF}\right)+\left(\overrightarrow{CF}-\overrightarrow{CD}\right)=\overrightarrow{ED}+\overrightarrow{FE}+\overrightarrow{DF}=\overrightarrow{DE}+\left(\overrightarrow{DF}+\overrightarrow{FE}\right)=\overrightarrow{ED}+\overrightarrow{DE}=\overrightarrow{EE}=\vec{0}$$
  Conclusion: $\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}-\overrightarrow{AE}-\overrightarrow{BF}-\overrightarrow{CD}=\vec{0}$
- Solution found by algorithms:
  Step 1:
  $$\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}-\overrightarrow{AE}-\overrightarrow{BF}-\overrightarrow{CD}$$
  $$=\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}+\overrightarrow{EA}+\overrightarrow{FF}+\overrightarrow{DC}\ (*)$$
  Step 2: Collect vectors can be applied by rules, so:
  $$(*)=\left(\overrightarrow{AD}+\overrightarrow{DC}\right)+\left(\overrightarrow{BE}+\overrightarrow{EA}\right)+\left(\overrightarrow{CF}+\overrightarrow{FB}\right)$$
  $$=\overrightarrow{AC}+\overrightarrow{BA}+\overrightarrow{CB}\ (**)$$
  Because it find rule to apply, do Step 2 again:
  We have:
  $$(**)=\left(\overrightarrow{AC}+\overrightarrow{CB}\right)+\overrightarrow{BA}=\overrightarrow{AB}+\overrightarrow{BA}\ (***)$$
  Do step 2 again: (***) $=\overrightarrow{AA}$
  Do step 2 again: $\overrightarrow{AA}=\vec{0}$.
  Conclusion: $\overrightarrow{AD}+\overrightarrow{BE}+\overrightarrow{CF}-\overrightarrow{AE}-\overrightarrow{BF}-\overrightarrow{CD}=\vec{0}$

2. Algorithms to Prove Vector Equality (Algorithms 2)

Vector expression which needs proving has form: L=R

- **Step 1**: Change expression L=R into form "$L=\vec{0}$ " (*), where: L=L-R
- **Step 2**: Simplify L by "Algorithms 1"
    - If $\vec{L}=\vec{0}$ (it means $(*) \Leftrightarrow \vec{0}=\vec{0}$) is right then stop and we have equality which needs proving is right.
    - Else go to step 3.
- **Step 3** : Check in the set of Facts to find fact which suit a rule in the set of Rules to create new fact
    - If it find rule can apply, then replace the rule into L and return step 2
    - Else stop and equality which need proving is false.

Illustration for "Algorithms to prove Vector equality":

Example: Let four points A, B, C, D, J. In which J is the midpoint AB. Proof: $\overrightarrow{JD}+\overrightarrow{JC}=\overrightarrow{AD}+\overrightarrow{BC}$

- Solution found by human:

  We have: $\overrightarrow{JD}+\overrightarrow{JC}=\overrightarrow{AD}+\overrightarrow{BC}$

  So: $(\overrightarrow{JD}+\overrightarrow{JC})-(\overrightarrow{AD}+\overrightarrow{BC})=\vec{0}$

  We have:

$$\begin{aligned} L &= (\overrightarrow{JD}+\overrightarrow{JC})-(\overrightarrow{AD}+\overrightarrow{BC}) \\ &= (\overrightarrow{JD}+\overrightarrow{JC})+(\overrightarrow{DA}+\overrightarrow{CB}) \\ &= (\overrightarrow{JD}+\overrightarrow{DA})+(\overrightarrow{JC}+\overrightarrow{CB}) \\ &= \overrightarrow{JA}+\overrightarrow{JB}\ (*) \end{aligned}$$

  Because J is midpoint of AB, then: $\overrightarrow{JA}+\overrightarrow{JB}=\vec{0}$

  So: (*) = $\vec{0}$

  Conclusion: $\overrightarrow{JD}+\overrightarrow{JC}=\overrightarrow{AD}+\overrightarrow{BC}$
- Solution found by algorithms:

  Step 1:

$$\begin{aligned} &\overrightarrow{JD}+\overrightarrow{JC}=\overrightarrow{AD}+\overrightarrow{BC} \\ &\Leftrightarrow (\overrightarrow{JD}+\overrightarrow{JC})-(\overrightarrow{AD}+\overrightarrow{BC})=\vec{0} \end{aligned}$$

  Step 2:

$$\begin{aligned} &\overrightarrow{JD}+\overrightarrow{JC}+\overrightarrow{DA}+\overrightarrow{CB}=\vec{0} \\ &\Leftrightarrow \overrightarrow{JB}+\overrightarrow{JA}=\vec{0}\ (*) \end{aligned}$$

  Step 3: We have the fact "J is midpoint of AB", check it in the set of Rules and we find a rule fixed: "*{J is midpoint of AB}* → *{$\overrightarrow{JB}+\overrightarrow{JA}=\vec{0}$}*". Then, replace this rule into the L of (*), where L= $\overrightarrow{JB}+\overrightarrow{JA}$. We have: $\vec{0}=\vec{0}$.

  Return to Step 2: Because $\vec{0}=\vec{0}$ is right, algorithms is stop.

  Conclusion: $\overrightarrow{JD}+\overrightarrow{JC}=\overrightarrow{AD}+\overrightarrow{BC}$

III. MODEL FOR ANALYTIC GEOMETRY

A. Model

In some knowledge domains, such as knowledge of Analytic Geometry, the knowledge can be represented by KBCO model (Knowledge Base of Computational Object) [2] without Operators and Hierarchical relation components. We'll call this model "KBCO model for Analytic Geometry". The structure of this model consists of 4 components:

(C, R, Funcs, Rules)

Where:

- ***C** is a set of concepts of computational objects (Com-object).* Consist of: "Point", "Vector", "Segment", "Angle", "Line", "Triangle", "Parallelogram", "Ellipse".
    - *Point* is a basis object.
    - *Segment*, Vector are objects of the first level.
    - *Angle, Line, Triangle, Parallelogram, Ellipse* are object of the second level.
- ***R** is a set of relations on the concepts.* Each relation is identified by <the name of relation> and the kinds of objects. Relation has one of following properties, such as: reflex, symmetry, bridge.

  Example: Relation "Slope" between a number and a line: [Slope,Real,Line],{}
- ***Funcs** is a set of functions.* Performing knowledge and rule of calculation in analytic geometry.

  Example: Function to calculate intersection between two lines: Point Intersection(Line,Line){" symmetry"}
- ***Rules** is a set of rules.* Almost properties, clauses, theorems in analytic geometry can be represented by rules on facts relating to Com-objects.

  Example: {va: Vector, vb: Vector, dt:Line, va "parallel" vb, va "parallel" dt]} → {vb parallel dt}

The kinds of facts in "KBCO model for Analytic Geometry" are as following:

- **Fact of kind 1**: information about object kind.
    - Structure: [<object>, <the kind of object>].
    - Example: [d,"Line"], [a,"Vector"].
- **Fact of kind 2**: a determination of an object or an attribute of an object.
    - Structure: <object> or <object>.<property>
    - Example: Segment[A,B], Segment[A,B].x
- **Fact of kind 3**: a determination of an object or an attribute of an object by a value or a constant expression.
    - Structure: <object> = <constant expression> or <object>.<property> = <constant expression>.
    - Example: a.x = 2, A=[1,2], a = [5,3]
- **Fact of kind 4**: equality on objects or attributes of objects.
    - Structure: <object>|<object>.<property> = <object>|<object>.<property>.
    - Example: O1.c = Segment[A,B], a = b, E.A1=[-a,0] ('a' can be object or property of object)
- **Fact of kind 5**: a dependence of an object or an attribute of an object on other objects by a formula.
    - Structure: <object> = <expression of objects or expression of properties of objects> or <object>.<property> = <expression of objects or expression of

properties of objects>
  - o Example: a = 2*c + b
- **Fact of kind 6**: a relation on objects or attributes of objects.
  - o Structure: [<the name of object>,<object1>,<object2>,…]
  - o Example: [“Belong”,M,d]
- **Fact of kind 7**: a determination of a function.
  - o Structure: <Function>
  - o Example: Vector(M,N)
- **Fact of kind 8**: a determination of a function by a constant expression.
  - o Structure: <function> = <constant expression>
  - o Example: Distance(M,d) = 3, Midpoint(A,B) = [1,2]
- **Fact of kind 9**: equality between an object or an attribute and a function.
  - o Structure: <object> = <function>
  - o Example: J = Midpoint(A,B)
- **Fact of kind 10**: equality between a function and other function.
  - o Structure: <function > = <function>
  - o Example: Midpoint(A,B) = Midpoint(M,N)
- **Fact of kind 11**: a dependence of a function on other functions by a formula.
  - o Structure: <function> = <expression of functions>
  - o Example: Vector(C,K)=2* Vector(A,K)

B. Organization of Knowledge Base

The knowledge base is organized by “KBCO model for Analytic Geometry” as the figure below:

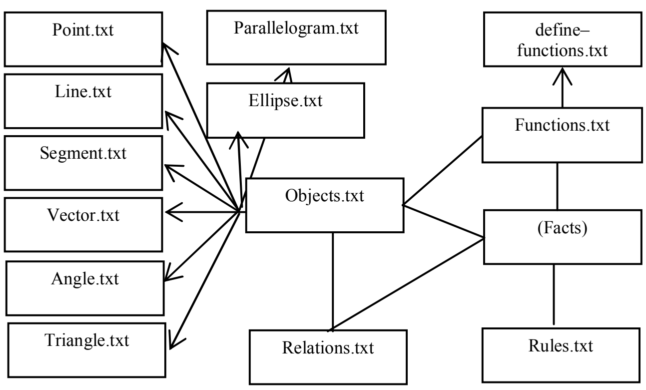

Fig. 2 Organization of Analytic Geometry knowledge domain

Where:

- The file “Objects.txt” stores names of concepts: Point, Segment, Line, Angle, Vector, Triangle, Parallelogram, Ellipse.
- The files <name of concept>.txt store the specifications of structures of Com-objects (there are 7 files are *“Point.txt”, “Segment.txt”,”Line.txt”,”Angle.txt”, “Vector.txt”, “Triangle.txt”, “Parallelogram.txt”, “Ellipse.txt*”)
- The file “*Relations.txt*” stores specification of deferent relations in Networks of Com-objects [2].
- The file “*Functions.txt*” stores the name of functions and “*define_functions.txt*” stores the specification of functions.
- The file “*Rules.txt*” stores deductive rules.

C. Model for Problems

We have studied and surveyed carefully on Analytic Geometry knowledge domain. We found out a model for those problems and it is able to specify for almost problems in Analytic Geometry high school education program. The structure of problem model is denoted by: **(O, M, Func, Facts, Functions, Goals)**

Where:

- **O** is a set of Com-objects of the problem.
- **M** is a set of attributes of objects in the set O.
- **Func** is a set of functions in the problem.
- **Facts** is a set of hypothetic facts of kinds from 1 to 6.
- **Functions** is a set of hypothetic facts of kind 7, 8, 9, 10, 11.
- **Goals** is a set of target facts of the problem.

Example: Give equation (d): x + 2y - 7 = 0.

a. Determine two points A, B which are intersection of Ox, Oy with (d).

b. Find out coordinate projection H of coordinate angle O on line (d).

c. Write equation (d’) which is symmetric with (d) through O.

The problem can be modeled by the following model:

- O = {d: Line, Ox: Line, Oy: Line, A: Point, B: Point, O: Point, H: Point, d’: Line}
- M= {d.ptdt, Ox.ptdt, Oy.ptdt, H.x, H.y}
- Func= {Intersection(Line, Line), Projection(Point, Line), Symmetry(Line, Point)}
- Facts = {d.ptdt: 3x + 4y -12 =0, Ox.ptdt: y=0, Oy.ptdt: x = 0, O(0,0)}
- Functions= {A = Intersection(d, Ox), B = Intersection(d, Oy), H = Projection(O, d), d’ = Symmetry(d, O)}
- Goals = {[A, B, H, (d’)]}

D. Algorithms

1. The Deductions in “KBCO Model for Analytic Geometry”

Step is the way to find out new facts from existing facts. It belongs to one of forms:

- Deduce_From3s: To find out new facts type 2 from facts type 3
- Deduce_From43s: To find out new facts type 3, 4 from facts type 3 and 4 by replacing variables in type 3 into type 4.
- Deduce_From53s: To find out new facts type 3, 4, 5 from facts type 3 and 5 by replacing variables in type

3 into type 5.

- Deduce_From45s: To find out new facts type 3 from facts type 4 and 5 by solving the equations system.
- Deduce_From8s: To find out new facts type 7 from facts type 8
- Deduce_From983s: To find out new facts type 3, 8 from facts type 3, 8 and 9 by replacing variables in type 8 (or fact type 3) into type 9.
- Deduce_Objects: To deduce and calculate inside the structure of each object. Every object take part in step have ability to perform fixed behavior to create new fact, to deduce and calculate on attributes of each object, own object or relative object which are establish base on object's foundation .
- Deduce_From9s: To find out new facts type 2, 3, 6, 7 and 8 from facts type 9 by calculating function.
- Deduce_Rules: To check rules can be applied
- Deduce_Funcs: To check functions can be applied
- Deduce_EqsGoal: To solve equations system includes n equations and n unknowns.

2. The Steps to Find out The Result

To find the solution of the problem, from hypothesis A, various kinds of reasoning can be applied to produce new facts and extend the set of defined variables until getting goal B of the problem. The basic idea is combining the reasoning process and some heuristics rules to improve the solving speed of the algorithm and get better solution.

The algorithm of "KBCO model for Analytic Geometry" consists of 9 steps which divided into 3 stages:

- **Stage 1**: Initiate
  - Step 1: Read and analyze hypothesis.
  - Step 2: Record the information of the problem by the Networks of Computational Objects.
  - Step 3: Initiate the empty solution and set initial states for some controlling variables.
- **Stage 2**: Deduce for finding the solution
  - Step 4: Check goal. If goal is obtained then goto step 9.
  - Step 5: Using heuristics rules to select a suitable solving step for producing new facts, new objects and getting the new state of reasoning process.
  - Step 6: If there is a solving step found in step 5 then record the information about the solving step, new objects in the set C and new facts in the current set of facts, and goto step 5.

  If goal is obtained then goto stage 3
  Else goto step 5.
  - Step 7: Else {selection in step 5 fails} Conclusion: Solution not found, and stop.
- **Stage 3**: Reduce the solution
  - Step 8: Reduce the solution found in stage 2 to get better solution. Starting from the target, we consider steps in solution to find which facts may deduce the goal and this fact will be considered as new targets for counting down in next step. This process becomes finish when all targets have been known or belong to hypothesis.
  - Step 9: Display the solution of the problem.

Example: Triangle ABC, M=[0,4] is midpoint of BC. Equation AC: x+4y-2=0, equation AB: 2x+y-11=0, straight line d through C, d: 2x+4y=0. Find the Area of triangle ABC.

- Summary:
  - Hypothesis: {M = Midpoint(B, C), M = [0, 4], d.f = (2*x+4*y = 0),Segment[A, B].f = (2*x+y-11 = 0),Segment[A, C].f = (x+4*y-2 = 0), [O1, "Triangle[A,B,C]"], ["Belong", C, d]}
  - Goal: {O1.S}
- Solution found by human:

  $A \in AB, A \in AC \Longrightarrow$A is root of equations system

  $$\begin{cases} 2x + y - 11 = 0 \\ x + 4y - 2 = 0 \end{cases} \Leftrightarrow \begin{cases} x = 6 \\ y = -1 \end{cases} \Leftrightarrow A = [6, -1]$$

  $C \in AC, C \in d \Longrightarrow$C is root of equations system

  $$\begin{cases} 2x + 4y = 0 \\ x + 4y - 2 = 0 \end{cases} \Leftrightarrow \begin{cases} x = -2 \\ y = 1 \end{cases} \Leftrightarrow C = [-2, 1]$$

  M is midpoint BC, we have M= [0,4], C=[-2,1] => B=[2,7]

  We have:

  $$S_{ABC} = \frac{1}{2}\left|(b_x - a_x)(c_y - a_y) - (c_x - a_x)(b_y - a_y)\right|$$

  $$= \frac{1}{2}|(2-6)(1+1) - (-2-6)(7+1)| = 28$$
- Solution using algorithms

  Apply algorithms through two stages (stage 1 and stage 2), we get a solution including 40 steps. After processing stage 3 that is used to reduce we have a solution including just16 steps:

  1. {AC.f=(x+4*y-2 = 0)} → { AC.f} by Deduce_From3s"
  2. {AB.f=(2*x+y-11=0)}→{AB.f}] by Deduce_From3s"
  3. {d.f = (2*x+4*y = 0)} → {d.f} by "Deduce_From3s"
  4. {M= Midpoint(B,C), M = [0, 4]} → {Midpoint(B,C) = [0, 4]} by "Deduce_From983s"
  5. {d.f} → {d} by "Deduce_ObjRules": "identify_object"
  6. {AB.f}→ {AB} by "Deduce_ObjRules": "identify_object"
  7. {AC.f}→{AC} by "Deduce_ObjRules": "identify_object"
  8. $\{C \in d$, C, $C \in AC\}$ → {C = Intersection(d,AC)} by "Deduce_Rules": "identify", {dt1: Line, dt2: Segment, W: Point, $W \in dt1, W \in dt2$ } → {W = Intersection(dt1,dt2)}
  9. {A$\in AB$, $A \in AC\}$ → {A = Intersection(AB,AC)} by "Deduce_Rules",

"identify", dt1: Segment, dt2: Segment, W: Point, $, W \in dt1, W \in dt2$ } → {W = Intersection(dt1,dt2)}

10. {C = [-2, 1]} →{ $x_C = -2, y_C = 1$} by "Deduce_From3s"
11. {A = [6, -1]} → {$,x_A = 6, y_A = -1$} by "Deduce_From3s"
12. {Midpoint (B,C) = [0, 4]} → {1/2* x_B -1 = 0, 1/2* y_B+1/2 = 4} by "Apply sk8": {Midpoint(B,C) = [0, 4]}
13. {1/2* x_B-1 = 0, 1/2* y_B+1/2 = 4, 2* x_B+ y_B-11 = 0} → {B = [2, 7]} by "Deduce_EqsGoal"
14. {B = [2, 7]} → { x_B = 2, y = 7} by "Deduce_From3s"
15. { x_A = 6, y_A = -1, x_B = 2, y_B = 7, x_C = -2, y_c= 1} → {O1.S = 28} by "Deduce_OConstructRela" {O1.S = 1/2*abs((x_B-x_A)*(y_C- y_A)-(x_C- x_A)*(y_B- y_A))}
16. {O1.S = 28} → {O1.S} by "Deduce_From3"

IV. Application

Depend on two models "KBCO model for Vector Algebra" and "KBCO model for Analytic Geometry", we developed two modules: module VectorAlgebraSolver to solve problems in Vector Algebra and module AnalyticGeometrySolver to solve problems in Analytic Geometry (those packages are written in Maple).

Combination Maple with C#: Process of making solution of a problem is designed and installed on Maple, but the solution is conventionalized in a defined data structure, which is difficult for users. To improve this problem, it needs a program having a visual interface and being easy to interact, but still reflect correctly the solution of problem. And C# is one of the powerful programming languages which can do that. All results and data structures received from the Maple solution are transferred to C# and use C# to demonstrate the solution in a natural way. Most of the Maple mathematical functions are written in Maple's language and translated by Maple's kernel which is written in C. Thus, combination C# with Maple isn't complex.

Two examples below present clearly a comparison between the solutions of the system and the solution of human:

Example for Vector Algebra: Give 4 points A, B, N, O. $\overrightarrow{NA} = -2\overrightarrow{NB}$. Proof: $\overrightarrow{OA} + 2\overrightarrow{OB} = 3\overrightarrow{ON}$

- Summary:
  - Hypothesis: {A,B,N,O: Point, $\overrightarrow{NA} = -2\overrightarrow{NB}$ }
  - Goal: {$\overrightarrow{OA} + 2\overrightarrow{OB} = 3\overrightarrow{ON}$}
- Solution found by human

  We have: $\overrightarrow{OA} + 2\overrightarrow{OB} = 3\overrightarrow{ON} \Leftrightarrow \overrightarrow{OA} + 2\overrightarrow{OB} - 3\overrightarrow{ON} = \overrightarrow{0}$

  We have: L = $\overrightarrow{OA} + 2\overrightarrow{OB} + 3\overrightarrow{NO}$ = $(\overrightarrow{NO} + \overrightarrow{OA})$ $+2\overrightarrow{OB} + 2\overrightarrow{NO}$ = $\overrightarrow{NA} + 2\overrightarrow{OB}$ + $2\overrightarrow{NO}$ (*)

  Because $\overrightarrow{NA} = -2\overrightarrow{NB}$

  So: (*) = $-2\overrightarrow{NB} + 2\overrightarrow{OB}$ + $2\overrightarrow{NO}$

  = $2\overrightarrow{BN} + 2\overrightarrow{OB}$ + $2\overrightarrow{NO}$ = $(\overrightarrow{BN} + \overrightarrow{OB} + \overrightarrow{NO})$

  = $2(\overrightarrow{ON} + \overrightarrow{NO}) = \overrightarrow{0}$ = R

  Conclusion: $\overrightarrow{OA} + 2\overrightarrow{OB} = 3\overrightarrow{ON}$
- Solution found by program:

  Step 1:

  L = $\overrightarrow{OA} + 2\overrightarrow{OB} - 3\overrightarrow{ON}$= $\overrightarrow{OA} + 2\overrightarrow{OB} + 3\overrightarrow{NO}$ = $(\overrightarrow{NO} + \overrightarrow{OA}) + 2\overrightarrow{NO} + 2\overrightarrow{OB}$ = $\overrightarrow{NA} + 2\overrightarrow{NO} + 2\overrightarrow{OB}$ = $\overrightarrow{NA} + 2(\overrightarrow{NO} + 2\overrightarrow{OB})$ = $\overrightarrow{NA} + 2\overrightarrow{NB}$

  Step 2:

  Because: $\overrightarrow{NA} = -2\overrightarrow{NB}$ So: $\overrightarrow{NA} + 2\overrightarrow{NB} = \vec{0}$

  So: L = $\vec{0}$

  Conclusion: $\overrightarrow{OA} + 2\overrightarrow{OB} = 3\overrightarrow{ON}$

Example for Analytic Geometry: Give 5 points A, B, C, E, F where A=[1,0], B=[-3,-5], C=[0,3], $\vec{c} = \overrightarrow{AF}, \vec{d} = \overrightarrow{CF}$ và $\overrightarrow{AE} = 2 * \overrightarrow{BC}, |\vec{c}| = |\vec{d}| = 5$. Find out coordinate H.

- Summary:
  - Hypothesis: {A = [1, 0], B = [-3, -5], C = [0, 3], c = Vector(A, F), d = Vector(C, F), c.length = 5, d.length = 5}
  - Goals: {F}
- Solution found by human

$$\vec{c} = \overrightarrow{AF} = (x_F - 1, y_F)$$
$$\vec{d} = \overrightarrow{CF} = (x_F, y_F - 3)$$
$$\begin{cases} |\vec{d}| = \sqrt{x_F^2 + (y_F - 3)^2} = 5 \\ |\vec{c}| = \sqrt{(x_F - 1)^2 + y_F^2} = 5 \end{cases}$$
$$\Leftrightarrow \begin{cases} x_F^2 + (y_F - 3)^2 = 25 \\ (x_F - 1)^2 + y_F^2 = 25 \end{cases}$$
$$\Leftrightarrow \begin{cases} x_F = -4 \\ y_F = 0 \end{cases} or \begin{cases} x_F = 5 \\ y_F = 3 \end{cases} \Leftrightarrow F = [-4,0], F = [5,3]$$

- Solution found by program:

1. A = [1, 0] → A by "Deduce_From3s"
2. C = [0, 3] → C by "Deduce_From3s"
3. c.length = $\sqrt{x_c^2 + y_c^2}$ → $\sqrt{x_c^2 + y_c^2}$ = 5 by "find_fact3_object" : c.length = 5
4. d.length = $\sqrt{x_d^2 + y_d^2}$ → $\sqrt{x_d^2 + y_d^2}$ = 5 by "find_fact3_object" : d.length = 5
5. A, F, c → x_c=x_F-1, y_c=y_F by "Apply sk9": c = $\overrightarrow{AF}$
6. C, F, d → x_d=x_F, y_d=y_F-3 by "Apply sk9": d = $\overrightarrow{CF}$
7. {x_c=x_F-1, y_c=y_F, x_d=x_F, y_d=y_F-3,$\sqrt{x_c^2 + y_c^2}$ = 5,$\sqrt{x_d^2 + y_d^2} = 5$} → F = [-4, 0] by "Deduce_EqsGoal"
8. {x_c=x_F-1, y_c=y_F, x_d=x_F, y_d=y_F-3,$\sqrt{x_c^2 + y_c^2}$ = 5,$\sqrt{x_d^2 + y_d^2} = 5$} → F = [5, 3] by "Deduce_EqsGoal"
9. F = [-4, 0], F = [5, 3] → F by "Deduce_From3s"

V. Conclusion

We had a thorough grasp of the organization, the store knowledge base, the model of problems, the algorithms and all of the design technique in KBCO model. After that, we set successfully two sub models of KBCO model for solving problems in Vector Algebra and Analytic Geometry knowledge.

The program was built in Maple, C# language. We have tested and evaluated seriously, the program gets following advantages:

+ The result is reasonable. Presenting the solution in detail, clearly through many steps close to human's thought.

+ The program's interface is clear and easy to use.

+ The result that we get in the experiment process is rather good. After comparing with the natural solutions, this program solves almost problems about Vector Algebra and Analytic Geometry.

Our purpose in future is to improve model, technique, and then develop program into complete computing software in the high Mathematics school support system. Besides the application was presented here, we think that these two sub models can be developed to apply for other field such as physics, chemistry.

REFERENCES

[1] Hoang Kiem & Do Van Nhon, Extension and Development of The Knowledge Models of Computational Objects, Proceedings of National Conference of some selective problems of Information Technology, Publisher of Science and Technology (2005).

[2] Nhon Van Do, Model for Knowledge bases of Computational Objects, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 8, May 2010

[3] Do Van Nhon, "A Program for Studying and Solving problems in Plane Geometry", in proceedings of International Conference on Artificial Intelligence 2000, Las Vegas, USA, 2000, pp. 1441-1447.

[4] Nhon Do Van, "A system that Supports Studying Knowledge and Solving of Analytic Geometry problems", 16th World Computer Congress 2000, Proceedings of Conference on Education Uses of Information and Communication Technologies, Beijing, China, 2000, pp. 236-239.

[5] Nhon Van Do & Tam Pham Huu, "The Extensive Computational Network and Applying in an Education Software", in proceedings of ICAIE 2009 – Wuhan, P.R China, August, 22-23, 2009, pp 720-727, ISBN 978-1-84626-010-0 (Volume 2).

[6] Tan Do Trung, Artificial Intelligence, Publisher of Education (1998)

[7] George F. Luger & William A Stubblefield, Artificial Intelligence, Addison Wesley Longman, Inc. (1998)

[8] Stuart Russell & Peter Norvig, Artificial Intelligence – A modern approach (second edition), Prentice Hall (2003)

[9] Nguyen Huu Khang – Using Mathematics software for Teaching and Researching science – University of Science Vietnam national university – 2004.

[10] PK Jain - Khail Ahmad, Textbook of Analytical Geometry of Two Dimensions(Second Edition), Published by New Age International (P) Ltd, ISBN: 0852264135, 2005

Nhon Do is currently a senior lecturer in the faculty of Computer Science at the University of Information Technology, Ho Chi Minh city, Vietnam. He got his MSc and Ph.D in 1996 and 2002 respectively, from The University of Natural Sciences – National University of Ho Chi Minh City. His research interests include Artificial Intelligence, Computer Science, and their practical applications, especially intelligent systems and knowledge base systems.

Diem Nguyen is an assistant professor in the faculty of Computer Science at the University of Information Technology, Ho Chi Minh city, Vietnam. She got her BS degree in 2011. Now she is studying master at The University of Information Technology – National University of Ho Chi Minh City. Her research interests are knowledge representation and automatic reasoning.

Trang Nguyen is an assistant professor in the faculty of Computer Science at the University of Information Technology, Ho Chi Minh city, Vietnam. She got her BS degree in 2011. Now she is studying master at The University of Information Technology – National University of Ho Chi Minh City. Her research interests are knowledge representation and automatic reasoning.