International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:6, 2010

Model for Knowledge Representation using Sample Problems and Designing a Program for Automatically Solving Algebraic Problems

Nhon Do, Hien Nguyen

***Abstract*—**Nowadays there are many methods for representing knowledge such as semantic network, neural network, and conceptual graphs. Nonetheless, these methods are not sufficiently efficient when applied to perform and deduce on knowledge domains about supporting in general education such as algebra, analysis or plane geometry. This leads to the introduction of computational network which is a useful tool for representation knowledge base, especially for computational knowledge, especially knowledge domain about general education. However, when dealing with a practical problem, we often do not immediately find a new solution, but we search related problems which have been solved before and then proposing an appropriate solution for the problem. Besides that, when finding related problems, we have to determine whether the result of them can be used to solve the practical problem or not. In this paper, the extension model of computational network has been presented. In this model, Sample Problems, which are related problems, will be used like the experience of human about practical problem, simulate the way of human thinking, and give the good solution for the practical problem faster and more effectively. This extension model is applied to construct an automatic system for solving algebraic problems in middle school.

***Keywords*—**educational software, artificial intelligence, knowledge base system, knowledge representation.

I. INTRODUCTION

IN the science of artificial intelligence, models and methods for knowledge representation play a very important role for designing knowledge base systems as well as expert systems. Nowadays there are many various knowledge models which have been suggested and applied. In the books [1], [2], [13] we already knew some common methods for knowledge representation in designing knowledge base systems. Nevertheless, those methods also have several drawbacks which make knowledge representation difficult to implement, especially for areas in general education such as plane geometry, analytic geometry, algebra, etc. Thus, the computational network has been introduced to be able to represent human's knowledge, especially for computational knowledge (see [8]). Furthermore, it has also been applied to construct supporting programs in education (see [7]).

However, in order to deal with a practical problem, G. Polya [11] pointed out that we first need to answer the following questions:

"- Have you seen it before? Or have you seen the same problem in a slightly different form?

- Do you know a related problem?
- Look at the unknown! And try to think of a familiar problem having the same or a similar unknown.
- Here is a problem related to yours and solved before. Could you use it? Could you use its result? Could you use its method? "

This corresponds to searching related problems which have been solved before, and then proposing an appropriate solution for the problem. Besides that, we can use the result of related problems for solving the problem. Such related problems are called *sample problems*. In this paper, we extend the computational network by adding sample problems component to the knowledge of the system. This extension allows us to represent human's thought about finding sample problems before by answering the following questions:

(1) Have I solved this problem before?
(2) This is a problem I solved before. Can I apply its result?

The extension model of computational networks was called *Computational Network using Sample Problems.* It is a necessary extension for modeling knowledge base. In addition, this extension model can be applied in constructing a system for automatically solving algebraic problems in middle school.

II. MODEL OF COMPUTATIONAL NETWORK USING SAMPLE PROBLEMS

A. *Computational Network:*

Definition 2.1: A *computational network* is a pair (M, R), in which $M = \{x_1, x_2, ..., x_n\}$ is a set of variables with simple values (or unstructured values), and $R = \{r_1, r_2, ..., r_m\}$ is a set of computational relations over the variables in the set M. Each computational relation $r \in R$ has the following form:

(i) An equation over some variables in M, or
(ii) Deductive rule $r : u(r) \rightarrow v(r)$, with $u(r) \subset M$, $v(r) \subset M$, and there are corresponding formulas to determine (or to compute) variables in v(r) from variables in u(r).We also define the set $M(r) = u(r) \cup v(r)$.

Remark: In many applications equations can be represented as deduction rules.

This model which was presented in [7], [8] was used in the design of many applications in education. However, the alternating current problems in general education program have specific characteristics so that a suitable model will be

Authors are with University of Information Technology,Vietnam.
e-mail: nhondv@uit.edu.vn

proposed to design a system to solve alternating current problems.

Definition 2.2: Given a computational net K = (M, R).
The *Goal* of problem has two parts:

("KEYWORD", ListObj)

Which
(1) "KEYWORD": is a word of the set { "PROVE", "COMPUTE", "SOLVE, "FIND"}
(2) ListObj: list of variables in M.
If KEYWORD is "PROVE" or "COMPUTE" or "SOLVE" then ListObj has only one element.
Else if KEYWORD is "FIND" then ListObj has three elements and
ListObj = ["key", x, y] in which "key" is a keyword of conditional, x is a parameter, y is a variable ($x \in M$, $y \in M$)
ListObj mean "*Find* x *for* y *satisfy* "key" *condition*"

Example 2.1:

a) "Prove: $(a+b)^2-(a-b)^2=4ab$"
Goal = ("PROVE", [$(a+b)^2-(a-b)^2=4ab$])

b) "Compute: $45*156728+55*156728$"
Goal = ("COMPUTE", [$45*156728+55*156728$])

c) "Solve: $(x+3)^4-2(x+3)^2+5=6$"
Goal = ("SOLVE", [$(x+3)^4-2(x+3)^2+5=6$])

d) "Find m for the equation
$$m.x^2-(2m+1)x+m^2=0$$
has no roots of x
"
Goal = ("FIND", ListObj)
ListObj = ["no roots", m , $m.x^2-(2m+1)x+m^2=0$]

Given a computational net (M, F). The popular problem arising from reality applications is that to find a solution to determine a set $H \subseteq M$ from a Goal which is specified by definition 2.2.

This problem is denoted by the symbol **(H, Goal)**,
H is the hypothesis and **Goal** is the goal of the problem.

Definition 2.3: Given a computational net K = (M, F).
(i) For each $A \subseteq M$ and $r \in R$, denote $r(A) = A \cup M(f)$ be the set obtained from A by applying f. Let $S = [r_1, r_2, ..., r_k]$ be a list consisting relations in F, the notation $S(A) = r_k(r_{k-1}(\dots r_2(r_1(A)) \dots))$ is used to denote the set of variables obtained from A by applying relations in S.
(ii) The list $S = [r_1, r_2, ..., r_k]$ is called a *solution* of the problem (H, Goal) ($H \subseteq M$) if S(H) satisfied Goal. Solution S is called a good solution if there is not a proper sublist S' of S such that S' is also a solution of the problem. The problem is *solvable* if there is a solution to solve it.

B. Computational network using sample problems model:

When dealing with a practical problem, a convenient way to proceed is considering whether we have met a similar or related problem before or not. If so, then the solution for the problem can be obtained effectively. Or we determine whether the result of related problems can be used to solve the practical problem or not. This leads to a requirement that model of knowledge base needs to be furnished a new component which can capture this behavior of human.

Definition 2.4: A *sample problem* has three components

(M_P, Goal , Sol)

Which:
(i) (M_p, Goal) is a problem of computational network
(ii) Sol: is a solution of problem (M_p, Goal)

Example 2.2:

Sample Problem for Solving the Quadric Equation

Object = $a.x^2+b.x+c$
We have rules for solving the quadric equation
Rule r_1: $\{\text{Object}\} \to \{a,b,c\} \quad (a \neq 0)$
Rule r_2: $\{a,b,c\} \to \{\text{delta} = b^2-4ac\}$
Rule r_3: Compare delta and number 0
Rule $r_{3.1}$: $\{\text{delta} < 0\} \to \{\text{Root} = \varnothing\}$
Rule $r_{3.2}$: $\{\text{delta} = 0\} \to \left\{\text{Root} = \left\{-\frac{b}{2a}\right\}\right\}$
Rule $r_{3.3}$: $\{\text{delta} > 0\} \to \left\{\text{Root} = \left\{\frac{-b \pm \sqrt{\text{delta}}}{2a}\right\}\right\}$
Solution of problem is Sol = $[r_1, r_2, r_3]$
Now, we have the sample problem S for solving the quadric equation as follow:
S = (M_p, Goal, Sol)
which: M_p = {Object}
Goal = ("SOLVE",[Object])
Sol = $[r_1, r_2, r_3]$

Definition 2.5: A *Computational network using sample problems (CNSP)* is a model which consists of three following components:

(M, Sample, R)

(1) **M** = $M_{num} \cup M_{func}$ is a set of attributes or objects, with simple valued (real or Boolean valued) or functional valued.
$M_{num} = \{O_1, O_2, O_3, \dots\}$ is the set of simple valued objects.
$M_{func} = \{f_1, f_2, f_3, \dots\}$ is the set of functional valued objects.
(2) **Sample** = $\{S_1, S_2, S_3, \dots\}$ is a set of sample problems.

(3) **R** = $R_{sample} \cup R_{knowledge} \cup R_{Heuristic}$ is a set of deduce rule, and R is the union of three subsets of rules R_{sample}, $R_{knowledge}$, $R_{Heuristic}$. Each rule r has the form r: u(r)→v(r), with u(r) is the hypotheses of r and v(r) is the conclusion of r. A rule is also one of the three cases below.

- Case 1: r ∈ R_{sample}. For this case, r is a rule for searching sample problem in **Sample**
- Case 2: r ∈ $R_{knowledge}$. For this case, r is a rule of knowledge domain.
- Case 3: r ∈ $R_{Heuristic}$. For this case, r is a heuristic rule of knowledge domain.

C. Application of model CNSP for representing algebra-knowledge base in middle school:

Knowledge domain about algebra in middle school can be modeling by model CNSP as followed:

1. **M** = $M_{num} \cup M_{func}$, which:

- M_{num} = {Real number; numeric expressions; Boolean expressions}

E.g.: "2", "3.8" , "4+5+6", "7*314+3*314", "1/2 < 3/4"

- M_{func} = {functional valued objects}

E.g.: $x^2, x^4+x^2+1,\ a^2-b^2,\ a^3+b^3,\ (y-1)^2 \geq 0$

2. **Sample** = {

- Sample problems about solving simple equation and quadric equation;
- Sample problems about solving simple inequations and quadric inequations;
- Sample problems about computing the numeric expressions reasonably;
- Etc.

}

3. **R** = $R_{sample} \cup R_{knowledge} \cup R_{Heuristic}$, which:

- R_{sample}: **s**et of **r**ules for searching sample problem in **Sample.**

E.g.:

o Rule for searching kind of Sample Problems

o Rule for comparison the goal of Sample problems

o Rule for determination the Sample Problem

- $R_{knowledge}$: **s**et of **r**ules in knowledge domain about algebra in middle school.

E.g.:

o $\_X^2 - \_Y^2 = (\_X + \_Y)(\_X - \_Y)$

o $\{\_X^2 \geq 0\} \Rightarrow \{\_X \text{ in } \mathbb{R}\}$

o $\{\_X \geq \_Y, \_Y \geq \_Z\} \Rightarrow \{\_X \geq \_Z\}$

o Rule for multiplying by 0, Rule for simplification expressions.

- $R_{Heuristic}$: **s**et of heuristic **r**ules for searching rules in $R_{knowledge}$.

E.g.:

o Heuristic rules for determination rule in $R_{knowledge}$

o Heuristic rules for application rules in $R_{knowledge}$

III. MODEL OF PROBLEMS AND ALGORITHMS

A. Model of problems:

Given a model Computational Network using Sample Problems (M,Sample,R). Model of problem in CNSP is denoted by the symbol:

(P, O, F) → Goal

Which

(1) **P** = {m_1, m_2, m_3, …} is a set of parameter in problem
(2) **O** = {O_1, O_2, O_3, …} is a set of Objects in problem
(3) **F** = {$fact_1$, $fact_2$, …} is a set of Facts in problem
(4) **Goal**: the purpose of problem, which is specified like define 2.3

Example 4.1: Find m for the equation has root of x

$$x^2 - 2(m-1)x + m^2 = 0$$

Model of this problem:

P:={m}

O:={ $x^2 - 2(m-1)x + m^2 = 0$ } ⊂ M_{func}

F:={}

Goal:=("FIND",["has root", m, $x^2 - 2(m-1)x + m^2 = 0$])

B. Algorithms:

The solution of problem (P,O,F)→Goal on model CNSP is defined similarly definition 2.3

Algorithm 3.1: Find a solution of problem (P,O,F)→Goal

Step 1: Solution ← empty;
  Solution_found ← false;
  H ← O ∪ P;
Step 2: Repeat
  Find a sample problem S
   (S is not founded before)
  if (S found) then
   begin
    H ← H ∪ S.Goal;
    Add S.Sol to Solution;
   end;
  if (Goal is satisfied) then
   Solution_found ← true;
 Until Solution_found **or** **not**(S found);
Step 3: **if** Solution_found **then**
  goto step 5;

```
        else
          goto step 4;
  Step 4: Repeat
      Hold ← H;
      Select r ∈ R_knowledge ∪ R_Heuristic;
      while not Solution_found and (r found) do
        if (applying r from H produces new facts)
        then
            begin
              H ← H ∪ M(r);
              Add r to Solution;
            end;
        if (Goal is satisfied) then
            Solution_found ← true;
        Select new r ∈ R_knowledge ∪ R_Heuristic;
      end while;
   Until Solution_found or (H = Hold);
  Step 5: if Solution_found then
            Solution is a solution of the problem;
         else
            There is no solution found;
```

Algorithm 3.2: Find a sample problem from problem (P,O,F)→Goal

```
  Step 1: SP ← Sample
          Sample_found← false
  Step 2: Repeat
         Select S in SP
         if kind of F = kind of S.M_p then
           begin
             if kind of Goal = kind of S.Goal then
                 Sample_found ← true
             Else if S.M_p ⊆ H then
                 Sample_found ← true
           end
         SP ← (SP – S)
    Until SP = {} or Sample_found
  Step 3: if Sample_found then
            S is a sample problem of the problem;
         else
            There is no sample problem found;
```

Algorithm 3.3: Find a good solution from a solution S = [r_1,r_2, ..., r_k] of the problem (P,O,F)→Goal on model CNSP (M,Sample,R).

```
  Step 1: NewS ← [];
          H ← O ∪ P;
          V ← Goal.ListObj;
  Step 2: for i := k downto 1 do
            if v(r_i) ∩ V ≠ ∅ then
              begin
                Insert r_i at the beginning of NewS;
                V ← (V – v(r_i)) ∪ (u(r_i) – H);
              end
  Step 3: NewS is a good solution.
```

IV. APPLICATION

Based on model CNSP and the architecture of a system for solving problems in [5], we have built the system that was called *"The automatic solving algebraic problems system"* by MAPLE and JAVA. Knowledge of system about algebra in middle school was represented by model CNSP as section II.C. The automatic solving algebraic problems system can solve algebraic problems in middle school automatically and the solution was more natural and closer with the way of thinking and solving of human. Moreover, this system was tested on all various kinds of algebraic exercises in Vietnam education about middle-school mathematics ([9]).

Some kinds of problem can be solved by system:

- Compute the numerical expression reasonably
- Solve the equation and basic inequalities
- Solve the system of equation in two variables
- Factorization of a Polynomial
- Simplification of algebraic expression
- Radical simplification
- Prove the equality and basic inequality

Furthermore, our system can be applied to solve problems with parameters such as: Find the parameters which satisfy some conditions of the problem, Solve equations and inequations based on parameters, etc.

Some examples below present clearly the effects of this system, and comparison between the solution of the system and the solution of the system human:

Example 4.1: Let x,y in R such as: $\frac{x}{3}=\frac{y}{7}$ and $2x-3y=16$. Find (x, y) by the rate formula.

Solution is given by our system:

From hypothesis, we have:

+ The rate: $\frac{x}{3}=\frac{y}{7}$

+ And the equation: $2x-3y=16$

We have:

$$\frac{x}{3}=\frac{2x}{6}$$

$$\frac{y}{7}=\frac{3y}{21}$$

Thus, by the rate formula, we have:

$$\frac{x}{3}=\frac{y}{7}=\frac{2x-3y}{6-21}=\frac{16}{-15}$$

So,

$$x = -\frac{16}{5}$$

$$y = -\frac{112}{15}$$

"

Example 4.2: Let the equation, with m is a parameter, and x is a variable:

$$x^2 - 2.m.x + 2m - 1 = 0$$

Find m such that this equation has at least a root of x.

Solution is given by our system:

"The quadratic equation:

$$x^2 - 2.m.x + 2m - 1 = 0$$

has coefficients:

$$a = 1 \qquad b = 2b' = 2m \qquad c = 2m - 1$$

We have:

$$\begin{aligned} \text{delta}' &= (b')^2 - a * c \\ &= (-m)^2 - (2m-1) \\ &= m^2 - 2m + 1 \end{aligned}$$

This equation has at least a root if and only if:

$$\text{delta}' \geq 0$$
$$\Leftrightarrow m^2 - 2m + 1 \geq 0$$
$$\Leftrightarrow (m-1)^2 \geq 0$$
$$\text{(true for all m in } \mathbb{R}\text{)}$$

So this equation has at least a root $\Leftrightarrow m \in \mathbb{R}$

"

Example 4.3: Let the equation, with m is a parameter, and x is a variable:

$$\left(m^2 - 4\right)x + 2 = m$$

Solve this equation by m.

Solution is given by our system:

"Solve the equation:

$$\left(m^2 - 4\right)x + 2 = m$$

$$\left(m^2 - 4\right)x = -2 + m$$

The coefficient of x has a set of roots:

$$\{-2 \quad , \quad 2\}$$

+ if parameter $m = -2$, then:

"This equation has no root"

+ if parameter $m = 2$, then:

"This equation has set of roots is the set of real numbers "

+ if parameter $m \notin \{-2 \ , \ 2\}$, then:

$$x = \frac{-2 + m}{m^2 - 4} = \frac{1}{m + 2}$$

V. CONCLUSION AND FUTURE WORK

Model of Computational network using Sample problems is the extension model of Computational Network. It is extended by adding the component of Sample Problems and improving deduction techniques on Computation Network. So, the processing of system become more intelligent and the resolution of practical problem is similar to human.

By model CNSP, the automatic solving algebraic problems system is designed successfully. The solution given by this system is natural, precise and has reasoning like human. Furthermore, this system has been estimated very well by the teachers and pupils used.

Compare the automatic solving algebraic problems system and **UMS** Software version 7.5 (**U**niversal **M**ath **S**olver, see [15]):

	UMS Software version 7.5	The automatic solving algebraic problems system
Same	Solve the kinds of problems which were presented in section IV with step-by-step solution.	
Differ-ence	In order to, this software can solve some kinds of problem about: • Plotting the graph • Complex number Besides the solution by textual, the UMS Software accompanies the solution with voice comments.	Our system can solve problems with parameters: • Solve equations and inequations based on parameters • Find the parameters which satisfy some conditions of the problem. These are high difficult problems which cannot be solved by usual systems. Moreover, the deduction of system is simulated the way of human thinking.

Besides that, ontology COKB-ONT which was presented in [3], [6] is a useful tool and method for designing practical knowledge bases, modeling complex problems, especially, it is a good method for modeling plane-geometry knowledge. In the future, we will also continue researching work to combine model CNSP and ontology COKB-ONT for designing the system that can solve the problems about plane geometry by using Sample Problems. With this combining, the system will better support for learning users and become a completely education program.

REFERENCES

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

[2] John F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks/Cole (2000).

[3] Do Van Nhon, *Constructing intelligent systems for computation – Research and development of knowledge representation models to design systems for automated solving problems*, PhD. thesis, National University of Ho Chi Minh City (2001-2002).

[4] 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).

[5] Do Van Nhon, *The architecture of a system for solving problems for learners and design techniques*, Scientific magazine of Education and Technology, Technical teachers' college of Ho Chi Minh City, No 2(4) 2007.

[6] Nhon Do, *An ontology for knowledge representation and Applications*, Proceeding of World Academy of science, engineer and technology, vol. 32, August 2008, ISSN: 2070-370.

[7] Nhon Van Do &Tam Pham Huu, *Extensive Computational Networks And Applying in an Educational Software,* Proceedings of 2009 International Conference on Artificial Intelligence and Education (ICAIE 2009), Wuhan, China, 2009.

[8] Nhon Van Do, *Computational Networks for Knowledge Representation*, World Academy of Science, Engineering and Technology, Volume 56, August 2009, ISSN 2070 – 3724 (ICCSISE 2009), Singapore, 2009.

[9] Vietnam Ministry of Education and Training, *Textbook and workbook of algebra in middle school*, Publisher of Education (2006-2007).

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

[11] G. Polya. *How to solve it*, Publisher of Education (1997).

[12] Lakemeyer, G. & Nebel, B. (1994), *Foundations of Knowledge representation and Reasoning*. Berlin Heidelberg: Springer-Verlag (1994).

[13] ToshinoriMunakata, *Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More*, Springer-Verlag London Limited (2008).

[14] Frank van Harmelem & Vladimir & Bruce, *Handbook of Knowledge Representation*, Elsevier (2008).

[15] http://www.umsolver.com/cgi-bin/club.pl?language=en