

PROBLEM SOLVING OF ELEMENTARY ALGEBRA BY HIERARCHICAL ABSTRACTION

Fujio Nishida, Yoneharu Fujita and Hiroji Kusaka
Department of Electrical Engineering,
Faculty of Engineering, University of Osaka Prefecture,
Sakai, Osaka, Japan 591

This paper presents a hierarchical problem-solving method which searches some applicable transformation rules to the precondition of the problem at various abstract levels using a tree-like address table and finds a global solving policy which reduces the difference between the current state and the goal. The frame of the problem is chosen from the field of elementary algebra.

1. INTRODUCTION

The abstraction of a problem and a relation which retains only their key characteristics will be indispensable for global solving of complicated problems. Many researches of mechanical problem solving using this concept have been actively done, but a lot of problems to be solved seem to remain for practical use.[1],[2]. This paper develops the method of EAPS(Elementary algebraic Problem Solver) of the previous paper[3] by adding (i)a tree-like address table of theorems and global relations for the efficient access to them and (ii)a mixed abstract notation of problems for flexible representation.

2. ABSTRACT EXPRESSIONS

In mechanical problem solving, it is considered indispensable to introduce an abstraction or an abbreviation method of expressions at least in complicated quantitative relations. Along this idea, we introduce an abstract form for quantitative terms and relations as follows;

idname-k(C-name:value,...,C-name:value), (1)
where 'idname' takes a term 't', a literal 'rq' or a set of literals ' $\overline{rq}$ '. k is the identification number though it is omitted in Table 1 for simplicity. The C-names('C-' is the abbreviation of the word 'characteristic' in this paper) are a variable 'var', a function class 'fc', a form of a literal 'form' and a sign of terms 'sgn'. The value of 'var' is a set of variables in an expression. The values of 'sgn' are {p,z,n} and the combinations of them. The item 'form' takes the values 0,1,2 and 3 for a solution form q(x, t(var:φ)), a semi-solution form q(y,t(var:{x})), a transposed form q(t(var:{x}),0) and the general form q(t(var:{x}),t(var:{x})) respectively. An abstract form eq.(1) is obtained by parsing an objective expression by the aid of the rewriting rules shown in Table 1.

```
t(var:φ) ::= const
t(var:{x},fc:(x:{fs})) ::= fs(x)
t(var:{x,y},fc:(x:{fs1,fs3})(y:{fs2,fs3}))
  ::= fs3(t(var:{x},fc:(x:{fs1})),t(var:{y},fc:
    (y:{fs2})))
 $\overline{t}$ (num:2,var:{x,y},fc:(x:{fs1})(y:{fs2}))
  ::= t(var:{x},fc:(x:{fs1})),t(var:{y},fc:(y:
    {fs2}))
q ::= =|>|<
rq(var:{x,y},fc:(x:{fs1})(y:{fs2}))
  ::= q(t(var:{x},fc:(x:{fs1})),t(var:{y},fc:(
    y:{fs2})))
 $\overline{rq}$ (num:2,var:{x,y}) ::= rq(var:{x}),rq(var:{y})
L ::=  $\bigwedge$  |  $\bigvee$  |  $\bigwedge \bigvee$ 
```

Table 1. A part of rewriting rules.

In Table 1, fs1, fs2 and fs3 are function symbols, $\overline{t}$ and $\overline{rq}$ denote sets of terms and literals respectively and num is the number of the elements of a set. The values of function classes about variables are polynomial 'poly', fractional 'fract' and so on. L is the abstraction of a compound logical operator.

The quantitative expressions in a given problem are reduced to three abstract forms; q(term,term), rq(C-name:value,...,C-name:value) and L(rq(C-name:value,...,C-name:value)). They are called the objective-based literal expression OBL, the characteristic-based literal expression CBL and the abstract compound-literals expression ACL. These three forms are constructed at the general identification and are permitted to appear in an expression as a mixed form.

In addition to the general identification, several special identification routines are implemented which identifies the C-value if the transformation rule to be applied contains a special kind of C-name such as the degree of a polynomial.

Problem descriptions take the following forms. Problems such as a proof-problem $(Qx)(P(x) \rightarrow R(x))$ are expressed as $((Qx)P(x) \parallel (Qx)R(x))$ based on the state space representation, where Qx is a sequence of quantifiers and both $P(x)$ and $R(x)$ are predicates and called the precondition and the goal respectively. Value-finding problems are also expressed in a similar format to proof problems as follows;

$((\exists x)P(x) \parallel (\exists x)Obj(L(x=t(var:\phi))))$, where ' $\parallel$ ' stands for an equivalence relation, and the goal part denotes to require an objective form corresponding to the ACL expression $L(x=t(var:\phi))$.

3. THEOREMS AND TRANSFORMATION RULES

Mathematical theorems are classified into several subsets by their characteristics and can be accessed by means of a tree-like address table shown in Fig.1.

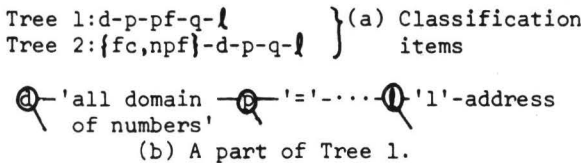


Fig.1 Address trees.

Tree 1 concerns primitive functions pf's such as addition, while Tree 2 concerns function classes fc's such as polynomials and also non-primitive functions npf's such as the absolute function. The C-item d means the domain where a theorem holds. p is the item that indicates an equality or inequality relation. p indicates the kind of quantifiers and l is the number of literals included in a premise of a theorem.

In addition to theorems, some relations called transformation rules 'TR's are introduced in order to find a global solving policy. There are two kinds of TRs, one of them is C-TRs and the other is P-TRs (i.e. problem-TRs). The form of a C-TR takes an implication form $(Qx)(r(x) \rightarrow s(x))$ (OP:S), where $r(x)$ and $s(x)$ are conjunctive normal forms of some abstract expressions of literals. The symbols ' $\underline{P}$ ' and ' $\underline{P}$ ' represent probable relations. The OP part indicates location areas of supporting theorems, TRs or procedures of the TR. Some of the TRs are shown in Table 2.

(1) Characteristic-TRs

- ① $rq(var:\{x\}, fc:poly)$
 $\Rightarrow L(Rq(var:\{x\}, fc:poly(deg:\{1,2\})))$ (OP:TR ②)
- ② $t(var:\{x\}, fc:poly)$
 $\Rightarrow \Pi(t(var:\{x\}, fc:poly(deg:\{1,2\})))$ (OP:PROC factorization)
- ③ $(\forall x) (\exists t(var:\forall u\{x\}, fc:poly(deg:(x:2))), 0)$

- ④ $\exists(\exists(\exists(\exists(sgn:pz, var:\bar{y})), 0))$ (OP:TH ①, TH ②)
- ⑤ $\exists(\exists(sgn:pz)) = t(sgn:pz)$
- (2) Problem-TRs
- ① $(T \parallel Q) \Leftrightarrow (Q \parallel T)$
- ② $(Q_1 \parallel Q_2) \Leftrightarrow (Q_1 \rightarrow Q_2 \parallel T)$

Table 2. Transformation rules.

The universal quantifier of TRs are omitted for simplicity. The logical connective symbol L, the predicate symbol q and the variable symbols x and y in the TRs are variables for which any corresponding instance or variable can be substituted. The predicate symbol q denotes equality or inequality. TR ① means that a polynomial relation will be reducible to a combined logical form which consists of several linear or quadratic relations. TR ② suggests that a polynomial term will be factorized into several linear or quadratic terms by a procedure. TR ③ shows an absolute inequality which contains quadratic terms about a variable can be transformed to a sum of definite sign terms such as definite positive terms, where TH ① and TH ② are

$$a^2 + a^2x^2 + b^2x + c = a^2(x+b/(2a))^2 + c - b^2/(4a)$$

and

$$a > 0 \wedge c - b^2/(4a) \geq 0$$

$$\Rightarrow (\forall x) a^2(x+b/(2a))^2 + c - b^2/(4a) \geq 0$$

respectively.

In P-TRs, the symbols Q, Q_1 and Q_2 are variables for which any logical formula can be substituted. The symbol T denotes a tautology such as $t(sgn:p) > 0$.

The access to TRs is performed by the aid of a tree-like address table a part of which is shown in Fig.2.

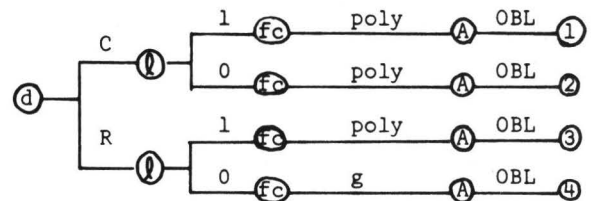


Fig.2. A part of address table of TRs.

In the address table, the items d, l and fc have the same meanings as in Fig.1. The value C denotes the complex number domain, while R denotes the real number domain. 0, 1 and 2 mean l=0, l=1 and l=2 respectively. A TR with l=0 is a no-premise relation. The value g of the item fc denotes the general-function class. 'A' indicates the abstraction levels of premises of TRs. If a TR involves several literal expressions of different abstraction levels, its address is recorded at the respective abstraction levels.

The following describes the main part of application procedures of TRs to the conjunctive nor-

mal form of the precondition of a problem. At the first step of application procedures, if the abstraction level of a TR is lower than that of a problem, the abstraction level of the TR is raised to that of the problem by the identification of the TR, otherwise the usual unification algorithm is adopted. Let $P_i \wedge \dots \wedge P_n \rightarrow Q \dots (a)$ and $P_i \wedge \dots \wedge P_m \rightarrow Q \dots (b)$ be a precondition of a problem and a TR respectively, where P' means an instantiation of P and $\{i_1, \dots, i_m\}$ is a subset of $\{1, \dots, n\}$, then the result of the application of (b) to (a) is $P_i \wedge \dots \wedge P_n \wedge Q'$. If the implication symbol is replaced by the equivalence symbol in eq. (b), $P_i, \dots, P_n$ are removed from the result.

4. SOLVING PROCEDURES

The solving procedures of EAPS are divided into three phases; (i) the identification of the current problem at the three abstract levels, ACL, CBL and OBL, (ii) the breadth-first search of a relevant global solving policies by the aid of the tree-like address table and (iii) the depth-first solving in a concrete form according to the suggestion of the obtained global solving policies. A part of a solving tree generated by EAPS is shown in Fig. 3, where the forms of PB, PB_i and PB_i' are $(P|G), (P|Q)$ and $(Q|G)$ respectively. The global solving policies are placed in the ascending order of the difference between the precondition and the goal of PB_i' which is evaluated based on the assigned values to the fundamental C-items.

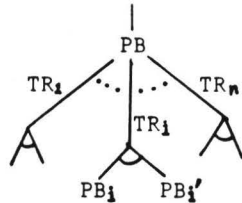


Fig. 3 A part of a solving tree.

If no global solving policy is found, EAPS searches applicable theorems by the aid of the tree-like address table of theorems. If EAPS fails to find applicable theorems, it applies P-TRs to the current problem then retries to solve the problem.

[Example] 'Prove that if k is a real number, the equation $x^2 - 2kx + 2k - 2 = 0$ of x has real roots'. The expression of the problem is

$$(k \in R) \mid (\exists x) \text{root}(\text{var}:x, \text{rel}:x^2 - 2kx + 2k - 2 = 0) \mid C \mid R \quad (1)$$

Since there is no applicable TR or TH to the precondition of eq.(1), the problem transformation is tried. The application of P-TR ① to eq.(1) yields

$$((\forall k)(\exists x)(\sim k \in R \vee \text{root}(\text{var}:x, \text{rel}:x^2 - 2kx + 2k - 2 = 0) \mid C \mid R) \mid T). \quad (2)$$

After finding no applicable TRs, one candidate of applicable theorems

$$(\exists x)(\text{root}(\text{var}:x, \text{rel}:a^2x^2 + b^2x + c = 0) \mid C \mid R) \mid T \mid b^2 - 4a^2c \geq 0$$

is found from the address area $fc:poly-d:real-p$:

$nq-q:V-1:1$ involved in the address table of THs. The application of the above theorem to the precondition of eq.(2) results in

$$((\forall k) \sim k \in R \vee k^2 - 2k + 2 \geq 0 \mid 1 \geq 0). \quad (3)$$

The CBL form of eq.(3) is

$$((\forall k) \sim k \in R \vee r_q - 3 \cdot 2(\text{var}:\{k\}, fc:poly) \mid r_q - 3 \cdot 3(\text{var}:\phi, form:2)). \quad (4)$$

The numbers 3.2 and 3.3 in eq.(4) denote the second and the third literals of eq.(3) respectively. The candidates of applicable TRs are ① and ③. The application of TR ① fails in the adoption of TR ② which is suggested by TR ①. The application of TR ③ to the OBL form of eq.(3) results in

$$((\forall k) \sim k \in R \vee \Sigma(\bar{\tau}(\text{sgn}:pz, \text{var}:\{k\}), 0) \mid \Sigma(\bar{\tau}(\text{sgn}:pz), 0)).$$

The application of the theorem TH ① suggested by TR ③ to eq.(3) leads to the result;

$$((\forall k) \sim k \in R \vee (k-1)^2 + 1 \geq 0 \mid 1 \geq 0). \quad (5)$$

The highest level of abstraction of applicable TRs to the precondition of eq.(5) is OBL, that is;

$$((\forall k) \sim k \in R \vee q - 5 \cdot 2(\Sigma(\bar{\tau}(\text{var}:\{k\}, fc:poly, \text{sgn}:pz), 0) \mid \Sigma(\bar{\tau}(\text{sgn}:pz), 0)). \quad (6)$$

The number 5.2 denotes the second literal of the precondition of eq.(5). The application of TR ④ to the precondition of eq.(6) yields

$$((\forall k) \sim k \in R \vee \Sigma(\bar{\tau}(\text{sgn}:pz), 0) \mid \Sigma(\bar{\tau}(\text{sgn}:pz), 0))$$

which leads to the completion of the proof.

6. CONCLUSION

The first version of EAPS[3] implemented on a mini-computer in an assembly language solved problems such as those which involve a simultaneous linear equation symbolically in about ten seconds. The new EAPS with additional facilities presented in this paper is under development on a computer with a 768 kB memory in LISP 1.5.

ACKNOWLEDGMENTS

The authors wish to thank Mr. K.Imori for his help in the implementation of EAPS.

REFERENCES

- [1] Sacerdoti, E.D. "Planning in a Hierarchy of Abstraction Spaces." *Artif. Intell.* 5:1 (1974) 115-135.
- [2] Bledsoe, W.W. "Non-resolution Theorem Proving." *Artif. Intell.* 9:1 (1977) 1-35.
- [3] Nishida, F. and Fujita, Y. "A Hierarchical and Heuristic Problem-solving in Elementary Algebra." In *Proc. UJCC-78*. UJCC Committee, San Francisco, California, October, 1978, pp. 99-103.