数理解析研究所講究録
511巻 1984年 26-48

Analogy and Generalization

Makoto Haraguchi

Research Institute of Fundamental Information Science
Kyushu University
Fukuoka 812, Japan

1. Introduction

Analogical reasoning (AR) is derivations of conclusions from premises like other reasoning. The premise in AR is a description that given two situations are similar in some respects, while the conclusion is one that the situations are similar in the other respects as well.

The analogical reasoning or analogical problem solving is so important in the studies of AI that many authors have investigated from various viewpoints [3,4,5,6,7]. However no mathematical framework for AR is found as far as the author's knowledge is concerned. So in the present paper, we try to build a mathematical theory for AR. We have paid our special attention to the following two observations which should be a basis of AR.

(1) We can utilize the past experiences to solve the current problem by detecting a similarity between the past and the current problems [3,4,5,6,7].

(2) We can acquire or learn a constraint and a general law by detecting the similarity and by identifying the similar parts of the two situations [5].

Thus the similarity detection is a key problem in AR. From this viewpoint, we take the first step toward a mathematical theory of analogy detection. The formalism is based on Winston's matching for analogy and is in terms of a first order language. The basic idea of our formalism is as follows.

(1) A similar part of two situations is a common part of them. Since the whole can deduce its part, we can refer to the similar part as a special kind of theorem derived from each of the situations.

(2) By taking logical NOT, we can view an analogy as a generalization by Plotkin[1,2]. Thus we can treat the analogy problem in nearly the same way as the generalization.

2. Extensible Relation Structures

Winston[5] describes a system which reasons and learns by analogy. His implemented system takes simple English like inputs which describe several facts about some situations such as Shakespeare's tragedies or scientific laws. Then the system translates them to kinds of networks called extensible relation representations. The extensible relation representation consists of situation parts as nodes that are tied together with relations. In order to express a supplementary description for the relation itself, a new kind of node called reference node is created. Such a node is hanging from the relation to which the node refers when we illustrate it as a figure. (See Example 1.1)

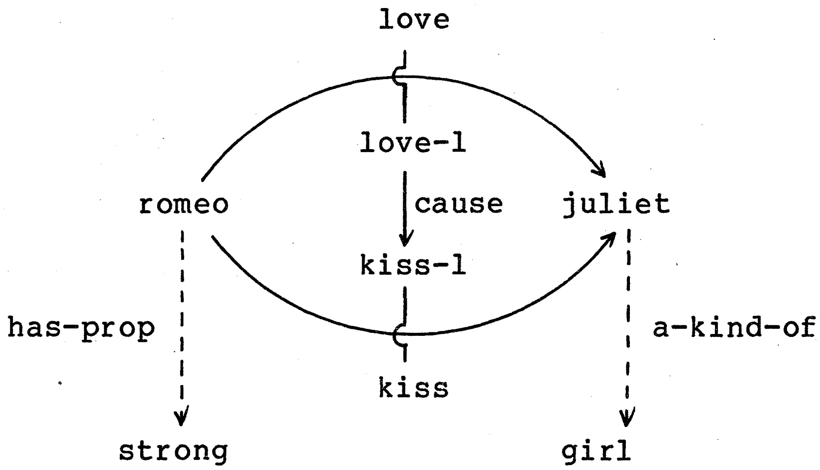
Then, for given two situations, matching for analogy is to pair off the situation parts, not reference nodes. In other word, the reference nodes are used only for describing various kinds of relations about situation parts. Therefore, a relational representation without the reference nodes is suitable when we use first order language. For this reason, we regard Winston's extensible relation representation as a general network structure S with pointer-type reference nodes, and then encode it to a set L(S) of ground atoms by eliminating the reference nodes. In the following definition, the situation parts and the reference nodes are simply called objects and references, respectively. Moreover internal properties of situation parts are defined by predicates over objects.

Definition 1.1 Extensible Relation Structure (ERS)

ERS S is a tuple (O,V,R,f,P), where V is a set of nodes called references, O is a set of nodes called objects, R is a family $\{R_i\}_{i \in I}$ of relations, each relation R_i has an arity $n > 1$ and is a subset of $(V \cup O)^n$, f is a one-to-one mapping $V \dashrightarrow \bigcup_{i \in I} R_i$ and is called reference relation, and P is a family of predicates over objects.

Before defining L(S), we encode S to a set C(S) of ground atoms with the references. In order to distinguish the usual relations from the reference relations, we use the reserved predicate symbols "refer" and "rel". Thus the relation names are treated as function symbols. The encoding is trivial, so we only show it by example.

Example 1.1 (Winston)



where "strong", "girl" are property's value, not situation parts (object), and love-1 and kiss-1 are refereces. As the result of coding, we have the set C(S) of ground atoms:

C(S) = {rel(love(romeo,juliet)), ... fact1
rel(kiss(romeo,juliet)), ... fact2
rel(cause(love-1,kiss-1)), ... fact3
refer(love-1,love(romeo,juliet)),
refer(kiss-1,kiss(romeo,juliet)),
has-prop-strong(romeo), ... fact4
a-kind-of-girl(juliet)} ... fact5.

Observe that love-1 refers the fact1 that romeo loves juliet, and kiss-1 the fact2. Since fact3 that love-1 cause kiss-1 is also a fact, an "extended" fact that cause(love(romeo,juliet),kiss(romeo,juliet)) also holds. Such a transitive derivation is easily described as follows:
For each function symbol r, we add the following rules to C(S).

transitive rule: $refer(X, r(X_1, \ldots, W, \ldots, X_n))$
$\quad \leftarrow refer(X, r(X_1, \ldots, X_j, \ldots, X_n)),$
$\quad\quad refer(X_j, W).$

terminal rule: $rel(r(X_1, \ldots, W, \ldots, X_n))$
$\quad \leftarrow rel(r(X_1, \ldots, X_j, \ldots, X_n)),$
$\quad\quad refer(X_j, W).$

Since the reference node refers to a particular instance of relation, we assume that for each reference v there exists exactly one atom refer(v,t) in C(S) for some term t. In general, if there are cyclic reference relations, then infinitely many true relations over objects and references are derivable as logical consequences. However, we have

Proposition 1.2

(1) For a reference v, there exists at most one term t which satisfies the condition C that refer(v,t) is logically implied by C(S) and has no references. Hence we can define "exp" as follows:

$$exp(v) = \begin{cases} v & \text{if } v \text{ is object,} \\ t & \text{if } v \text{ satisfies the condition C,} \\ \omega & \text{if otherwise} \end{cases}$$

(2) Let L(S) be the set of all ground atoms which are logically implied by C(S) and has no reference symbols. Then L(S) is only finite.

Since L(S) is only finite , we chose L(S) as a logical representation of ERS S.

3. Matching for Analogy

G. Polya[9] said as follows:

"Two systems are analogous, if they agree in clearly definable relations of their respective parts."

Then the analogy detection (or matching for analogy) is to find the correspondences of the parts with the agreements on the definable relations. In order to deal with the problem of analogy detection computationally, we must effectively decide if the relations agree or not. For this reason, we simply consider that two relations r and r' agree if they have the same relation name. This restriction of the agreements is just the same as that of Winston in which he call the agreements of relations "evidences". In this section, we formally define the evidences using ERS and characterize them in terms of first order language.

Definition 3.1

(1) Given two ERSs $S_i = (O_i, V_i, R_i, f_i, P_i)$ (i=1,2) a pairing Φ is a one-to-one relation over O_1 and O_2.

(2) For given pairing $\Phi \subseteq O_1 \times O_2$, we extend Φ to Φ^+ which

pairs reference nodes also. Φ^+ is defined by $\bigcup_n \Phi(n)$, where

$$\Phi(0) = \Phi,$$

$$\Phi(n+1) = \Phi(n) \cup \{\langle u,u'\rangle \in V_1 \times V_2 : \langle a,a'\rangle,\langle b,b'\rangle \in \Phi(n),\ f_1(u)=r(a,b), \text{ and } f_2(u')=r(a',b')\}$$

Proposition 3.2

$\langle u,v\rangle \in \Phi^+$ iff $\omega \neq exp(u) = exp(v)$ (Φ), where $t=t'(\Phi)$ means identity that term t and t' are exactly the same by identifying each a and b with $\langle a,b\rangle$ in Φ.

Now define evidences.

Evidence I: Paired objects $\langle a,a'\rangle \in \Phi$ have the same property; that is p(a) and p(a') hold for some $p \in P_1 \cap P_2$.

Evidence II: Paired nodes $\langle u_i,v_i\rangle \in \Phi^+$ (i=1,n) are in the same relationship; that is, $r(u_1,\ldots,u_n) \in R_1$ and $r(v_1,\ldots,v_n) \in R_2$ hold for some $r \in R_1 \cap R_2$.

By using Proposition 3.2, we have:

Proposition 3.3

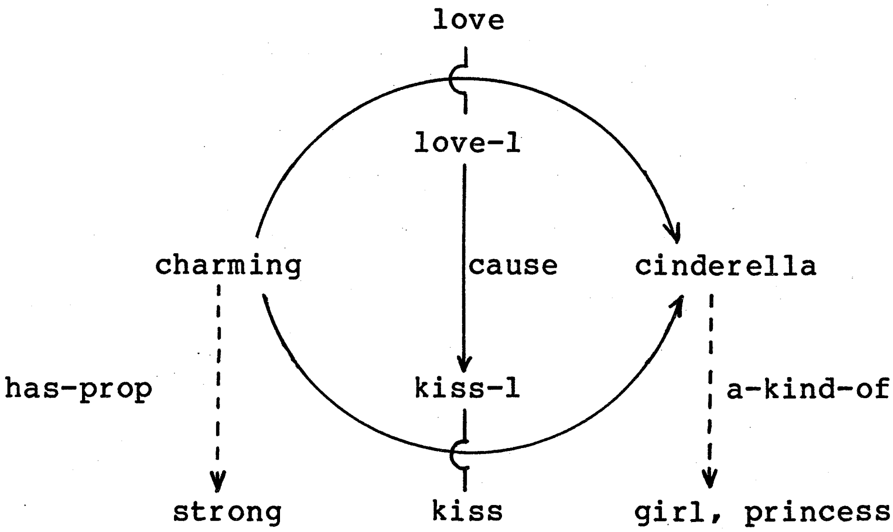
Evidence I and II are stated as follows:

Evidence I: For $p \in P_1 \cap P_2$, $\langle a,a'\rangle \in \Phi$, both p(a) and p(a') holds; that is $p(a) \in L(S_1)$ and $p(a') \in L(S_2)$.

Evidence II: For $rel(t_i) \in L(S_i)$ (i=1,2), $t_1 = t_2$ (Φ) holds for some relation r.

Example 3.3

One of situations is shown in Example 2. . Another is the following:



For pairing = {<romeo,charming>,<juliet,cinderella>}, we have five evidences:

<has-prop-strong(romeo), has-prop-strong(charming)>,

<a-kind-of-girl(juliet),a-kind-of-girl(cinderella)>,

<rel(love(remeo,juliet)),rel(love(charming,cinderella))>,

<rel(kiss(romeo,juliet)),rel(kiss(charming,cinderella))>,

<rel(cause(love(romeo,juliet),kiss(romeo,juliet))),
rel(cause(love(charming,cinderella),
kiss(charming,cinderella)))>.

4. Analogy Theorems

According to Proposition 3.3, an evidence can be a pair of identical atoms $<A_1,A_2> \in L(S_1) \times L(S_2)$, where the identity means $A_1 = A_2$ (Φ). Note that this atom pairing is one-to-one.

Let E(Φ) be the set of all evidences as pairs of atoms , and p_i be the usual projection. Since p_1(E(Φ)) and p_2(E(Φ)) are completely identical subsets of $L(S_i)$, E(Φ) represents a common identical parts of situations when S_i represents some situation. Moreover, we can represent this partial identity by the corresponding formula.

First assign variable $X_{<a,a'>}$ for each $<a,a'> \in \Phi$. For each evidences in E(Φ), we associate an atom A_i as follows: Atom $p(X_{<a,a'>})$ with evidence <p(a),p(a')>, and an atom $rel(term(X_{<a_1,a'_1>},\ldots,X_{<a_n,a'_n>}))$ with $<rel(term(a_1,\ldots,a_n)),rel(term(a'_1,\ldots,a'_n))>$. Then the desired formula is:

$$W(\ \Phi\ ) :\ X_{<a_1,a'_1>},\ldots,\ X_{<a_n,a'_n>}[A_1 \wedge \ \ldots\ \wedge A_k].$$

Clearly $L(S_i)$ ---> W(Φ) is valid (i=1,2), and the pairing is conversely computed as an pair of answer substitution of resolution proof. In fact,

$\sigma_1 = \{X_{<a,a'>}/a\}$, $\sigma_2 = \{X_{<a,a'>}/a'\}$ are

answer substitutions of $C(S_i)$ ---> W(Φ), and the

$\{<X\sigma_1, X\sigma_2> : X \in Var(W(\ \Phi\ )\}$ is Φ.

36

Since $W(\Phi)$ can define the partial identity of objects and atoms, we will call W an analogy theorem. Now we give a formalism of analogy.

Let S_1 and S_2 be conjunctive set of ground atoms with no common individual constants. Let $\mathbf{L}=\mathbf{L}[S_1;S_2]$ be the set of formulas W such that W is a conjunctive set of atoms, all variables are existensially quantified, and S_i ---> W is valid for i=1,2.

The following proposition is due to Skolemization.

Proposition 4.1 (Duality)

Let $A_1,\dots,A_n$ be atomic formulas. Then

$W= \exists x_1,\dots,\exists x_n[A_1 \wedge \dots \wedge A_m]$ ---> $W'=\exists y_1,\dots,\exists y_m[B_1 \wedge \dots \wedge B_n]$ is valid iff $\overline{B}_1 \vee \dots \vee \overline{B}_n$ subsumes $\overline{A}_1 \vee \dots \vee \overline{A}_m$.

Especially, when every A_i is ground atom, if σ is an answer substitution of W then $\overline{B}_1 \vee \dots \vee \overline{B}_n$ σ -subsumes $\overline{A}_1 \vee \dots \vee \overline{A}_m$ and vice versa.

From the proposition 4.1, we identify W in $\mathbf{L}$ with a clause C in $\mathbf{C}= \mathbf{C}[\overline{S}_1;\overline{S}_2]$ by taking NOT-operation, where $\mathbf{C}$ is the set of all clauses that subsumes both of $\overline{S}_i$. We call this identification as a duality.

Since the pairing Φ and the corresponding atom pairing are one-to-one, we constraints W in **L** this one-to-one condition called Partial Identity Condition (PIC). The term "partial identity" is due to Klix[8].

Let σ be a pair of substitution $(\sigma_1; \sigma_2)$. For variable or atom A, $\langle A\sigma_1, A\sigma_2\rangle$ is denoted by $A\sigma$. For set W of variables or atoms, $\{A\sigma : A \in W\}$ is denoted by $W\sigma$. Note that, for a conjunctive set W of atoms, $W \in \mathbf{L}[S_1;S_2]$ iff there exists a pair σ of substitutions such that $W\sigma \subseteq S_1 \times S_2$.

Definition 4.2

(1) A pair of substitution σ is said to satisfy PIC for W in **L** if $W\sigma \subseteq S_1 \times S_2$, and both $W\sigma$ and $Var(W)\sigma$ are one-to-one relations.

(2) For given S_i, we say W in **L** an analogy theorem under S_1 and S_2 if there exists a pair of substisutions $\sigma = (\sigma_1; \sigma_2)$ which satisfies PIC for W. In this case we call (W, σ) an analogy, and $W\sigma$ the evidence set of the analogy (W, σ).

5. Analogies and Implications

For given S_1 and S_2, there are many analogy theorems and analogies. Then the most important problem is to determine if one analogy is better than another. That is, the problem of ordering of analogies. In the previous studies of analogy, this ordering is done by using numerical score [4,5]. However the maximal score is unique, but, analogies with the maximal score are not unique. For this reason, we should have a more structural ordering.

Definition 5.1 (Plotkin[1,2])

For given clauses C_1 and C_2, if a clause C σ_i-subsumes C_i (i=1,2), then C is called a generalization of C_i (i=1,2). In this case, we call a pair (C,σ) a generalization diagram of C_1 and C_2, where σ is a pair (σ_1, σ_2).

generalization diagram

$$\begin{tikzcd} C_1 & & C_2 \\ & C \arrow[ul, "\sigma_1"] \arrow[ur, "\sigma_2"'] & \end{tikzcd}$$

According to the duality stated in Section 4, we have

(1) $(W = \exists x_1,\ldots,\exists x_m[A_1 \wedge \ldots \wedge A_n], \sigma)$ is an analogy under S_i

iff (2) $(\overline{W}, \sigma)$ is a generalization diagram of $\overline{S_1}$ and $\overline{S_2}$ which satisties PIC

iff (3) $(W' = \{A_1,\ldots,A_n\}, \sigma)$ is a generalization diagram of S_i which satisfies PIC.

In what follows, we use (3) as the notation of analogies.

Plotkin also defined a quasi-order and an equivalence of generalizations of C_i. By his definition, the equivalent generalizations are logically equivalent. However, in the case of analogy, logical equivalence does not means an equivalence of analogy. For instance, consider the following example:

Example 5.2

$$
\begin{tikzcd}
S = \{p(f(a)),p(b)\}. & & S' = \{p(f(a')),p(b')\} \\
& W = \{p(f(X)),p(Y)\} \arrow[ul, "{X/a,\ Y/b}"'] \arrow[ur, "{X/a',\ Y/b'}"] & \\
& W' = \{p(f(A))\} \arrow[u, "A/X"'] \arrow[uul] \arrow[uur] &
\end{tikzcd}
$$

are two generalization diagrams as analogies. Observe that W and W' are logically equivalent, and that evidence set of W' is a proper subset of that of W. Moreover, the object that the variable A denotes in W' is obtained by substituting A by the variable X in W. This means that the above diagram commutes.

In general, for given two analogies (A, σ) and (B, τ), when there exists a subsumption η that makes the corresponding generalization diagrams commutative, then the partial identity

$X\tau$ for variable X in B is described by the partial identity $X\eta\sigma$. Thus the condition of diagram commutativity is a syntactic way of describing one analogy B by another A. Of cource, B is worse (or weak) than A.

Definition 5.3

For given S_i (i=1,2), an analogy (A, σ) is called better than an analogy (B, τ), if there exists a substitution such that B η -subsumes A and that $X\tau_i = X\eta\sigma_i$ for all variable X in B (i=1,2). In this case we denote (A, σ) > (B, τ). we also define (A, σ) ~ (B, τ) if (A, σ) $\geq$ (B, τ) and vice versa.

Proposition 5.4

(1) If (A, σ) $\geq$ (B, τ) then $A\sigma \supseteq B\tau$.

(2) The ordering $\geq$ for analogies is a quasi-order.

6. Canonical Analogy Structures

The purpose of this section is to establish a canonical search space for analogy. For the detection of analogy , if we have object-pairing first and then show its evidences, then the corresponding method for analogy detection is essentially bottom-up, and the search space is the set of all object pairs.

However, like the bottom up method, a top-down detection of analogy is also possible. In this case we must find a partial identity P of ground atoms, and then establish object pairings which have their evidence sets as a part of P. The search space introduced in this section is for top-down analogy detection, and is called Canonical Analogy Structure (CAS) for given P.

Definition 6.1 Let S_i be sets of ground atoms,

(1) A pair of atoms is called compatible if they have the same predicate.

(2) $S \subseteq S_1 \times S_2$ is called a selection if S is one-to-one.

(3) For a given selection $S = \{\langle A_i, B_i \rangle : 1 \leq i \leq n\}$, a poset $(D(S), \leq)$ is defined as follows:

(3-1) D(S) consists of all equivalence classes [L] of generalizations of literals $tup(A_1,\ldots,A_n)$ and $tup(B_1,\ldots,B_n)$, where "tup" is a new predicate symbol to denote a tuple, the generalizations of literals are similarily defined by regarding literals as unit clauses, and the equivalence of literals is define by alphabetical variance.

(3-2) $[L_1] \leq [L_2]$ iff L_1 subsumes L_2 as unit clauses.

Note that when we use the above tuple notation, the predicate symbols appearing in A or B are treated as function symbols. In what follows, equivalent class [L] is represented by its arbitrary element L.

Example 6.2 D(S) for S={<p(f(a)),p(f(a'))>,<p(b),p(b')>}

$$
\begin{tikzcd}
\text{tup(p(f(a)),p(b))} & & \text{tup(p(f(a')),p(b'))} & \\
& \text{tup(p(f(X)),p(Y)) ...... top} \arrow[ul, dashed] \arrow[ur, dashed] & & \\
\text{tup(p(f(X)),Z)} \arrow[ur] & & \text{tup(p(X),p(Y))} \arrow[ul] & \\
& \text{tup(p(X),Z)} \arrow[ul] \arrow[ur] & & \text{tup(Z,p(X))} \arrow[ul] \\
& & \text{tup(X,Y) ...... bottom} \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Note that D(S) is a finite lattice with least generalization as $\cap$ and unification as $\cup$.

The aim of introducing D(S) is to extract analogies which have their evidence sets as subsets of S. For T=tup(e_1,...,e_n) D(S), if e_j is a variable then e_j plays no role for object pairing. Therefore we ignore such a e_j. d(T) is defined by { $e_{i,1}$,...,$e_{i,k}$}, where $e_{i,j}$ are all non-variable arguments. Now we define the canonical structure for given selection.

Definition 6.3 CAS[S] is an ordered structure (V,$\leq$) such that
(1) V consists of all (d(T), σ) such that

$T \in D(S)$, $\sigma = (\sigma_1, \sigma_2)$,

$T\sigma_1$=tup($A_1, \ldots, A_n$),

$T\sigma_2$=tupe($B_1, \ldots, B_n$), and

σ satisfies PIC.

(2) $\leq$ is the quasi-order of analogies.

Proposition 6.4

(1) If $(W, \sigma) \in C(S)$ then (W, σ) is an analogy under $p_i(S)$.

(2) CAS(S) is a poset. That is the quasi-order becomes a partial order by restricting analogies to those of CAS(S).

The reason why the author call CAS(S) "canonical" is due to the following lemma.

Lemma 6.5

Let (W, τ) be an analogy under S_1 and S_2 with its evidence set $W\tau \subseteq S \subseteq S_1 \times S_2$. Then there exists $(d(T), \sigma) \in$ CAS(S) such that $(d(T), \sigma) \geq (W, \tau)$.

7. Maximal Analogies

The purpose of matching for analogy is to extract the common identical parts as large as possible. That is, we must find maximal common identical subsets of S_i. This maximality is naturally defined by using the quasi-order of analogies, and we give a procedure which search all the maximal analogies.

Definition 7.1

(1) An analogy (W, σ) is called maximal if $(W, \sigma) \sim (W', \tau)$ holds whenever $(W', \tau) \geq (W, \sigma)$.

(2) A selection S is called maximal if there exists no selection such that $S' \supsetneq S$.

Proposition 7.2

If (W, σ) is a maximal analogy, then there exists an equivalent maximal element $(d(T), \tau) \in CAS(S)$ for some maximal selection S.

The inverse of Proposition 7.2 does not hold in general.

Example 7.3 Consider the following non-extensible relation structure.

$$\begin{tikzcd} a \arrow[r, "rel_1"] \arrow[d, "rel_2"'] & b \\ c \arrow[r, "rel_1"'] & d \arrow[u, "rel_3"'] \end{tikzcd} \qquad \begin{tikzcd} a' \arrow[d, "rel_2"'] & b' \arrow[l, "rel_1"'] \\ c' \arrow[r, "rel_1"'] & d' \arrow[u, "rel_3"'] \end{tikzcd}$$

Note that the above two figures represent

$S_1=\{rel_1(a,b),rel_2(a,c),rel_1(c,d),rel_3(d,b)\}$ and

$S_2=\{rel_1(b',a'),rel_2(a',c'),rel_1(c',d'),rel_3(d',b')\}$,

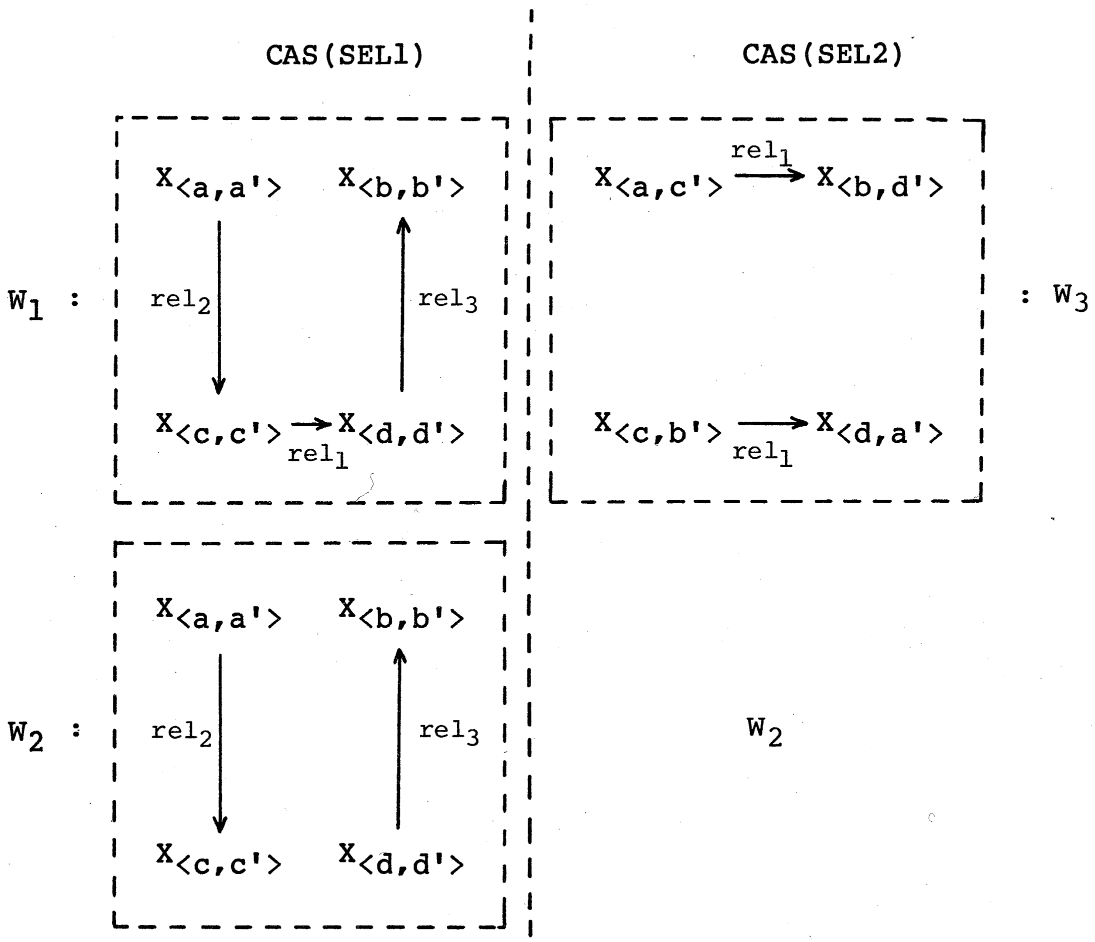
respectively.

Then there are two maximal selections SEL1 and SEL2.

$SEL1=\{<rel_1(a,b),rel_1(b',a')>,<rel_2(a,c),rel_2(a',c')>,$
$\quad <rel_1(c,d),rel_1(c',d')>,<rel_3(d,b),rel_3(d',b')>\}$

$SEL2=\{<rel_1(a,b),rel_1(c',d')>,rel_2(a,c),rel_2(a',c')>,$
$\quad <rel_1(c,d),rel_1(b'.a')>,rel_3(d,b),rel_3(d',b')>\}$

The parts of CAS(SEL1) and CAS(SEL2) are:



Observe that W_1 is maximal in CAS(SEL1), W_2 and W_3 are maximal in CAS(SEL2), and that W_2 is not maximal in CAS(SEL1).

By summarizing the above obsevation,

Theorem 7.4

Let S be a maximal selection, and (W, σ) be a maximal element of CAS(S). Then (W, σ) is not maximal analogy, iff there exists maximal selection $S' \supseteq W\sigma$ such that (W, σ) is not maximal element in CAS(S').

According to Proposition 7.2 and Theorem 7.4, we can get all the maximal analogies (up to equivalence for analogies) in the following way:

Compute the maximal elements in CAS(S) for maximal selction S, and check the condition of Theorem 7.4.

Note that, in order to compute maximal elements in CAS(S), it suffices to search D(S) downward from the top (which is the least generalization) until an element T satisfies PIC. Then the corresponding $(d(T), \sigma)$ is the desired maximal element.

8. Concluding Remarks

We have given some formal definitions of the extensible relation representation and the matching for analogy, and then we have formalized the problem of analogy detection which plays a key role in analogical reasoning. There still remain many problems to be solved:

(1) Analogy as a pairing of function symbols. Note that in this paper we have dealt with the analogy as a pairing of ground terms.

(2) Use of deduction or abstraction. In order to extract useful analogy, we should positively make use of information about situations, which are related to the use of deduction. In fact, Winston[5] said: "Some deduction or abstraction are necessary before matching."

Owing to limited space, we omit all the proofs. For details, see [10].

ACKNOWLEDGEMENTS

The author wishes to acknowledge Professor S. Arikawa for his valuable suggestions and encouragement.

References

[1] Plotkin, G.D. (1970), A note on inductive generalization, Machine Intelligence 5, 153-216.

[2] Plotkin, G.D. (1971), A further note on inductive

generalization, Machine Intelligence 6, 101-124.

[3] Chapman, D. (1982), A programming testing assistant, CACM, 25,9, 625-634.

[4] Tangwongsan, S and Fu, K.S. (1979), An application of learning to robot planning, Int. J. of Computer and Information Sciences, 8,4, 303-333.

[5] Winston, P.H. (1980), Learning and reasoning by analogy, CACM, 23,12,689-703.

[6] Carbonell, J.G. (1981), A Computational model of analogical problem solving, IJCAI-81,147-152.

[7] Kling, E,K (1971), A paradigm for reasoning by analogy, Artificial Intelligence, 2, 147-178.

[8] Klix, R.E. and Meer, F.D. (1979), Analogical reasoning - an approach to mechanisms underlying human intelligence performances, Human and Artificial Intelligence, North-Holland.

[9] Polya, G. (1954), Induction and analogy in mathematics, Princeton University Press.

[10] Haraguchi, M. (in prep.), Toward a mathematical theory of analogy detection.