ANALOGICAL REASONING USING TRANSFORMATIONS OF RULES

Haraguchi, Makoto
Research Institute of Fundamental Information Science, Kyushu University

https://doi.org/10.5109/13372

出版情報：Bulletin of informatics and cybernetics. 22 (1/2), pp.1-8, 1986-03. 統計科学研究会
バージョン：
権利関係：

Bulletin of Informatics and Cybernetics Vol. 22, No. 1~2, 1986

ANALOGICAL REASONING USING TRANSFORMATIONS OF RULES

By

Makoto HARAGUCHI*

Abstract

A formalism of analogical reasoning is presented. The analogical reasoning can be considered as a deduction with a function of transforming logical rules. From this viewpoint, the reasoning is defined in terms of deduction, and is therefore realized in a logic programming system. The reasoning system is described as an extension of Prolog interpreter.

1. Introduction

Analogical reasoning is an important reasoning method in our problem solving activities. Based on some analogy between two domains, we often transform knowledges of one domain into those of another, and make use of the transformed knowledges to solve the problem and to reason some unknown facts. Generally the reasoned facts are not necessarily true, however, the analogy under which the facts are reasoned gives an evidence showing that the facts may be true. Some computational methods for analogical reasoning have been discussed from various viewpoints [1, 2, 5, 6, 7, 11, 12, 13]. However, due to lack of mathematical formalism, neither methods to compare various analogies nor power of analogical reasoning have been described. The purpose of this paper is to present a mathematical formalism of analogical reasoning and a system which performs the analogical reasoning. Since the knowledges are generally represented by formulas, it is natural to consider the domains as axiom systems. However, in the present paper, we assume that the knowledges are simply facts about individuals in the domains and that the domains consist of such facts.

In order to formalize the analogical reasoning, we need, first of all, the definition of analogy. Since deduction is of importance to the analogical reasoning, it is desired that one language can describe them. From this viewpoint, Haraguchi [3] has defined the analogy in terms of first order language. We briefly review the definition of analogy in 2. Secondly we need to show a system which performs the analogical reasoning. We consider in this paper Winston's analogy-based reasoning [13] which makes use of causal structures of facts. Regarding causal structures as logical rules, we view the analogy-based reasoning as a deduction with a function of transforming rules. The transformation of rules is the key notion introduced in this paper, and is defined in Section 3. Then, using the transformation of rules, we define a set of facts

* Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan

which are conclusions of analogical reasoning. This set is called a conclusion set. In Section 4, we show a backward reasoning method which derives the facts in the conclusion set. The method is described as an extension of Prolog interpreter which tries to "prove" the facts in the conclusion set by finding possible transformations of rules.

2. Analogy as a Partial Identity

We start with reviewing some necessary definition, stated in [1], to precisely discuss the analogical reasoning. The analogy we consider is based on Polya's clarified analogy [3]:

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

Here each system consists of a set of parts and a set of relations between the parts. Thus when we say that some systems are analogous each other, they have at least one pair of relations, one from each, which agree under a correspondence of parts.

Each relation in the system can be represented as a ground atom, and the system is described by a set of ground atoms. For example, a family is a system of relations such as *parent*(a, b) and *father*(b, c), where "parent" and "father" are predicate symbols, and a, b and c are constant symbols to denote the members of family as system parts. Now consider two analogous systems S_1 and S_2:

$$S_1=\{A_1, \cdots, A_n\},$$

$$S_2=\{B_1, \cdots, B_m\},$$

where A_i and B_j are ground stoms representing relations in the systems.

Since S_1 and S_2 are analogous, there exists a pair

$$\langle A_i, B_j\rangle \in S_1 \times S_2$$

of relations which agree each other. We say that two relations agree if they have the same relation name (i.e. predicate symbol). Thus the pair $\langle A_i, B_j\rangle$ is written as

$$\langle p(t_1, \cdots, t_k), p(t'_1, \cdots, t'_k)\rangle,$$

where the pairing $\{\langle t_i, t_i'\rangle\}$ of terms is the correspondence under which A_i and B_j agree. We also say that these ground atoms are identified under the pairing of terms. Generally there may exist several pairs of atoms identified under a pairing of terms. These pairs of identified atoms form a set

$$\{\langle p_1(t_{11}, \cdots, t_{1n(1)}, p_1(t'_{11}, \cdots, t'_{1n(1)})\rangle,$$
$$\vdots$$
$$\langle p_k(t_{k1}, \cdots, t_{kn(k)}), p_k(t'_{k1}, \cdots, t'_{kn(k)})\rangle\}$$

with a pairing of terms

$$\{\langle t_{ij}, t'_{ij}\rangle : 1 \leqq j \leqq n(i),\ 1 \leqq i \leqq k\}.$$

Introducing a variable X_{ij} for each pair $\langle t_{ij}, t'_{ij}\rangle$, we represent this set by the following set W of atoms and pair $u=\langle u_1, u_2\rangle$ of substitutions.

$$W=\{p_1(X_{11}, \cdots, X_{1n(1)}), \cdots, p_k(X_{k1}, \cdots, X_{kn(k)})\},$$

$$u_1=\{X_{ij}\leftarrow t_{ij} : 1\leqq j\leqq n(i),\ 1\leqq i\leqq k\},$$

$$u_2=\{X_{ij}\leftarrow t'_{ij} : 1\leqq j\leqq n(i),\ 1\leqq i\leqq k\}.$$

The variable X_{ij} represents the pair

$$X_{ij}=\langle X_{ij}u_1,\ X_{ij}u_2\rangle=\langle t_{ij},\ t'_{ij}\rangle.$$

We sometimes write the pair of substitutions as $X_{ij}-\langle t_{ij},\ t'_{ij}\rangle$. Moreover each atom A in W represents the pair $Au=\langle Au_1,\ Au_2\rangle$. Hence $Var(W)u=\{Xu : X\in Var(W)\}$ is the pairing of terms, and $Wu=\{Au : A\in W\}$ is the set of identified atoms, where $Var(W)$ denotes the set of all variables in W. We require that both $Var(W)u$ and Wu are one-to-one. Now we can give the definition of formal analogy.

DEFINITION 1. Let S_1 and S_2 be finite sets of ground atoms with no common individual constants, and let u be a pair $\langle u_1,\ u_2\rangle$ of substitutions. Then we say that u satisfies the *partial identity condition* for a set W of atoms, if $Wu\subseteq S_1\times S_2$, and both Wu and $Var(W)u$ are one-to-one relations. In this case, we call $(W,\ u)$ an *analogy* between S_1 and S_2.

We have defined an analogy between two sets of ground atoms. Each ground atom represents a relation between system parts. Definition 1 still stands to define an analogy concerning more complex relations, such as relations about relations. (For details, see [3].)

3. Transformation of Rules Based on Analogy

As mentioned in the introduction, we consider Winston's analogy-based reasoning [13]. Since the power of analogy-based reasoning depends on the ability of deduction, and since the underlying principle of analogy-based reasoning can be viewed as the rule transformation defined in this section, we propose to deal with analogical reasoning in a deductive system with the function of transforming rules.

Winston [12, 13] has asserted that causal relations enable a common-sense reasoning that reasons some relations in a system from a relation in another system, provided that the two systems are analogous. His assertion is based on the principle that it seems reasonable for the common reasons to lead to a common effect. The principle is depicted in Fig. 1, where the relation $r_1(a,\ b)$ and $r_1(a',\ b')$ are identified by an analogy

$$(W,\ u)=(\{r_1(X,\ Y)\},\ \{X-\langle a,\ a'\rangle,\ Y-\langle b,\ b'\rangle\}),$$

and the cause relation

$$r_1(a,\ b)\Longrightarrow r_2(c,\ d)$$

shows that $r_1(a,\ b)$ works as a reason for $r_2(c,\ d)$ to hold. For the system S_2, $r_1(a',\ b')$ holds. We regard $r_1(a,\ b)$ and $r_1(a',\ b')$ as a common reason, since they are identified by the analogy $(W,\ u)$. Then the principle suggests that a common effect corresponding to $r_2(c,\ d)$ may hold in S_2. The common effect should be identified with $r_2(c,\ d)$ by

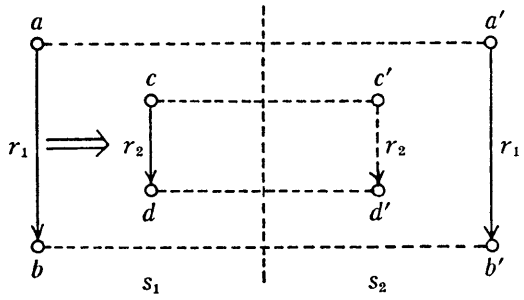

Fig. 1. The principle of analogical reasoning.

(W, u). Hence it must be $r_2(c', d')$ which is a missing relation in S_2.

Here we should notice that the cause relation works just like a logical rule. In order to reason the missing relation $r_2(c', d')$, we may just apply the logical rule

$$R' : r_2(c', d') \longleftarrow r_1(a', b'),$$

which is equal to

$$R : r_2(c, d) \longleftarrow r_1(a, b)$$

except for the pairing $Var(W)u$, to the relation $r_1(a', b')$ which already holds in S_2. In other words, the principle of analogical reasoning can be realized in a deductive system with the function of transforming rule R to R', because the missing relation $r_2(c', d')$ is deduced by applying modus ponens to R' and $r_1(a', b')$.

The transformation is now formally defined in terms of Definition 1. In what follows, we assume that each cause relation of the reasons $A_1, \cdots, A_k$ and the effect B is encoded to a definite clause $B \leftarrow A_1, \cdots, A_k$, and we simply call it a rule.

DEFINITION 2. Let (W, u) and $C_1 : A \leftarrow A_1, \cdots, A_k$ be an analogy and a ground rule, respectively. Then we say that a ground rule $C_2 : B \leftarrow B_1, \cdots, B_k$ is a *C_1-analogue* under (W, u), if there exists a definite clause $C : D \leftarrow D_1, \cdots, D_k$ such that $Cu_i = C_i (i = 1, 2)$ and $\{D, D_1, \cdots, D_k\} \subseteq W$ hold.

We call the conversion from C_1 to C_2 a transformation under (W, u). The clause C in Definition 2 is called a generalization of C_1 and C_2 ([8, 9]). In what follows, we represent the transformation by the following schema:

$$\frac{C_1 : A \longleftarrow A_1, \cdots, A_k}{C_2 : B \longleftarrow B_1, \cdots, B_k},$$

where the dotted line shows that the upper rule is transformed into the lower rule.

According to Definition 2, the principle of analogical reasoning is now represented by the following schema:

$$\frac{\dfrac{A \longleftarrow A_1, \cdots, A_k}{B \longleftarrow B_1, \cdots, B_k} \qquad B_1, \cdots, B_k}{B},$$

where the real line shows modus ponens. Thus the analogical reasoning is a combination of the usual deduction and the rule transformation. This schema is called a fundamental schema.

Generally reasoning is a process of applying inference rules to derive some facts. Hence it is natural to consider a process in which the rule transformation and modus ponens are applied consecutively. For instance, consider the following derivation of facts, where A_i and B_j are facts in S_1 and S_2, respectively.

$$\dfrac{\dfrac{A_1 \longleftarrow A_2 \qquad \dfrac{A_2 \longleftarrow A_3 \qquad \dfrac{B_2 \longleftarrow B_3 \qquad B_3}{B_2}}{B_1 \longleftarrow B_2}}{B_1}}$$

It should be noticed that the premise B_2 to conclude B_1 is derived by the fundamental schema. Hence we derive the fact B_1 by applying the fundamental schema two times.

In the derivation above, we transform rules in S_1 into those in S_2. It is also natural to transform rules in S_2 to derive some facts in S_1. For instance, we allow a derivation shown in the following.

$$\dfrac{\dfrac{B_1 \longleftarrow B_2 \qquad \dfrac{A_2 \longleftarrow A_3 \qquad \dfrac{B_2 \longleftarrow B_3 \qquad B_3}{A_2}}{A_1 \longleftarrow A_2}}{A_1}}$$

Now we give a formalism of analogical reasoning as a reasoning system which has two inference rules: modus ponens and transformation of rules.

DEFINITION 3. Let S_i be a set of facts (ground atoms) and ground rules for $i=1, 2$. Then we define the set S_i^+ of facts as follows:

$$S_i^+ = \cup S_i^n,$$

$$S_i^0 = \{A : S_i \vdash A\},$$

$$S_i^{n+1} = S_i^n \cup \{A : S_i^n \cup R_i^n \vdash A\},$$

$$R_i^n = \{C : B \leftarrow B_1, \cdots, B_k :$$
there exists a rule C': $A \leftarrow A_1, \cdots, A_k$ *in* S_j $(i \neq j)$, *and an analogy* (W, u) *between* S_1^n *and* S_2^n *such that* C *is a* C'*-analogue under* (W, u).$\}$

The set S_i^+ is called a conclusion set.

Note that S_i^n is the set of ground atoms which are derived by transforming rules at most n times. We should also notice that the set of analogies (W, u) changes as n increases.

4. Reasoning Method

In this section, we present a Prolog program *reason* which tries to "prove" facts in the conclusion sets. The program *reason* itself is an extension of Prolog interpreter. Hence we may consider that *reason* has goals to be established. As such goals, we consider two types of goals meaning the questions

"Is A in S_1^+?" and "Is B in S_2^+?".

We represent these goals by

$$fact_1(A) \quad \text{and} \quad fact_2(B),$$

respectively. Similarly we represent a rule $A \leftarrow B_1, \cdots, B_k$ in S_i by a clause

$$fact_i(A) \longleftarrow fact_i(B_1), \cdots, fact_i(B_k).$$

As a Prolog program, *reason* has the following properties:

(1) A fact B is in S_i^+ if *reason* ($fact_i(B)$) succeeds, and

(2) A fact B is not in S_i^+ if *reason* ($fact_i(B)$) fails.

From the definition, it is clear that a fact A is in S_i^+ iff A is derived by a finite number of applications of modus ponens and the rule transformation. Prolog interpreter realizes the application of modus ponens. Hence, first of all, *reason* is designed to be an extension of Prolog interpreter. First three clauses of *reason* interprets pure-Prolog programs:

(*C*1) *reason*(*true*) :— !.

(*C*2) *reason*((*Goal*, *Goals*)) :— !, *reason*(*Goal*), *reason*(*Goals*).

(*C*3) *reason*(*Goal*) :—*clause*(*Goal*, *Goals*), *reason*(*Goals*).

Now we show a method to realize the function of transforming rules. For this purpose, we exemplify how *reason* transforms rules. Consider the following sets of facts and ground rules.

$$S_1 = \{g(a, c),\ f(a, d),\ p(d, c),$$
$$p(d, c) \longleftarrow f(a, d),\ g(a, c)\},$$
$$S_2 = \{f(a', d'),\ g(a', c'),$$
$$r(c', d') \longleftarrow p(d', c')\}.$$

To show that $r(c, d)$ is in S_1^+, reason has the following goal:

$$\longleftarrow fact_1(r(c, d)).$$

This goal never succeeds as long as we use *C*1, *C*2 and *C*3, since $r(c, d) \notin S_1$. So we must try to transform some rule in S_2. The new clause *C*4 performs the act of transforming rule, and is applied when *C*1, *C*2 and *C*3 fails to be applied

(*C*4) *reason*(*Goal*): —

prematch(*Goal*, *Tgoal*, *Tgoals*),
transform(*Goal*, *Tgoal*, *Tgoals*, *Fgoals*, *Agoal*),
reason(*Tgoals*),
reason (*Fgoals*),
call(*Agoal*).

Given a goal in S_i, the predicate *prematch* in the body of *C*4 tries to find a possible rule in $S_j (i \neq j)$ to be transformed, and returns a rule in S_j whose head and body are *Tgoal* and *Tgoals*, respectively. For the goal $fact_1(r(c, d))$, *prematch* succeeds and returns the rule $R_1 : r(c', d') \leftarrow p(d', c')$ in S_2 to be transformed.

The predicate *transform* in the body of *C*4 tries to transform this rule for $fact_1$ ($r(c, d)$). Firstly *transform* generalizes $r(c, d)$ and $r(c', d')$, and obtains a pairing $\{\langle c, c'\rangle, \langle d, d'\rangle\}$. Then, using this pairing, *transform* applies the pairing to $p(d', c')$, the body

of R_1. Since c' and d' are paired with c and d, respectively, we have $p(d, c)$ as the body of R_1-analogue. As a result, *transform* obtains the following transformation:

$$\frac{r(c', d') \longleftarrow p(d', c')}{r(c, d) \longleftarrow p(d, c).}$$

From the Definition 3, we must show both $p(d, c) \in S_1^+$ and $p(d', c') \in S_2^+$. Hence *reason* has the new goals:

$$\longleftarrow fact_2(p(d', c')),\ fact_1(p(d, c)).$$

For each goal, *reason* is applied recursively. First we apply *reason* to $fact_2(p(d', c'))$, and then $fact_1(p(d, c))$.

Since $p(d', c') \notin S_2^0$, we try to transform

$$R_2 : p(d, c) \longleftarrow f(a, d),\ g(a, c)$$

in S_1 for $p(d', c')$. Generalizing $p(d, c)$ and $p(d', c')$, we have the pairing $\{\langle c, c'\rangle, \langle d, d'\rangle\}$. This pairing defines no term to be paired with the constant a appearing in the body of R_2. For this reason, we introduce a variable X_a for a, and consider the following rule transformation:

$$\frac{p(d, c) \longleftarrow f(a, d),\ g(a, c)}{p(d', c') \longleftarrow f(X_a, d'),\ g(X_a, c')}$$

Then *reason* has the new goals to be established:

$$\longleftarrow fact_1(f(a, d)),\ fact_1(g(a, c)),$$
$$fact_2(f(X_a, d')),\ fact_2(g(X_a, c')),$$

It should be noticed that the value substituted to the variable X_a must not violate the partial identity condition (Definition 1), since we define the rule transformation under some analogy (Definition 2). In other words, we must verify that the pairing $\{\langle c, c'\rangle, \langle d, d'\rangle, \langle a, X_a\rangle\}$ is one-to-one. For this purpose, *transform* returns the predicate instance

$$Agoal : analogy\ ([[c, c'], [d, d'], [a, X_a]]),$$

where *analogy* is a predicate to verify if the argument list $[[c, c'], [d, d'], [a, X_a]]$ is one-to-one or not. Then *Agoal* is called when the derived goals succeed and the variable X_a is instantiated to some ground term. For the goal

$$\longleftarrow fact_1(f(a, d)),\ fact_1(g(a, c)),$$
$$fact_2(f(X_a, d')),\ fact_2(g(X_a, c')),$$

we have the answer substitution with $X_a = a'$. Moreover

$$analogy\ ([[d, d'], [c, c'], [a, a']])$$

succeeds. Hence we have shown $p(d', c') \in S_2^+$.

It remains to show $fact_1(p(d, c))$. Clearly $fact_1(p(d, c))$ succeeds, since $p(d, c) \in S_1$. As a result, we have $r(c, d) \in S_1^+$. We have also shown that *reason* ($fact_1$ ($r(c, d)$)) succeeds when *reason* is interpreted by Prolog interpreter.

Generally some transformation of rule R may fail to derive some facts. In such a case, reason must try to transform another rule R' alternative to R. However, using Prolog's backtracking, we can perform this act of searching alternative transformations. In fact, Prolog goes back and attempts to re-satisfy

prematch (Goal, Tgoal, Tgoals).

Re-satisfying prematch, we have another rule R' to be transformed.

5. Concluding Remarks

Based on the formalism of analogy, we have succeeded to treat analogical reasoning formally. As a result, we obtain a method to realize the analogical reasoning in a logic programming system. The implementation of the method has been developed for DCL U-station, and is written by CProlog.

The analogy treated in the present paper is a "syntactic" one. Extending the notion of analogy, we are preparing a theory of analogy in which we can consider a semantic aspect of analogical reasoning.

Acknowledgement

The author is grateful to Prof. Setsuo Arikawa for valuable discussions.

References

[1] CARBONELL, J.G.: *A Computational Model of Analogical Problem Solving*, IJCAI-81 (1981), 147-152.
[2] CHAPMAN, D.: *A Programming Testing Assistant*, Comm. ACM, **25**, (1982), 625-634.
[3] HARAGUCHI, M.: *Towards a Mathematical Theory of Analogy*, Bull. of Inform. Cybernetics, **21**, (1985), 29-56.
[4] HARAGUCHI, M. and ARIKAWA, S.: *Analogical Reasoning based on the Theory of Analogy*, Res. Rept. Res. Inst. Fund. Inform. Sci., Kyushu Univ., No. 105, (1985).
[5] KLING, R.E.: *A Paradigm for Reasoning by Analogy*, Artificial Intelligence, **2**, (1971), 147-178.
[6] KLIX, R.E. and MEER, F.D.: *Analogical Reasoning-An Approach to Mechanisms Underlying Human Intelligence Performance*, Human and Artificial Intelligence, North-Holland, (1979).
[7] PLAISTED, D.A.: *Theorem Proving with Abstraction*, Artificial Intelligence, **16**, (1981), 47-108.
[8] PLOTKIN, G.D.: *A Note on Inductive Generalization*, Machine Intelligence **5**, 153-216.
[9] PLOTKIN G.D.: *A Further Note on Inductive Generalization*, Machine Intelligence **6**, (1971), 101-124.
[10] POLYA, G.: *Induction and Analogy in Mathematics*, Princeton University Press, (1954).
[11] TANGWONGSAN, S. and FU, K.S.: *An Application of Learning to Robot Planning*, J. Computer and Inform. Science, **8**, (1979), 303-333.
[12] WINSTON, P.H.: *Learning and Reasoning by Analogy*, Comm. ACM, **23**, (1980), 689-703.
[13] WINSTON, P.H.: *Learning New Principles from Precedents and Exercises*, Artificial Intelligence, **19**, (1983), 321-350.

Communicated by S. Arikawa
Received August 29, 1985