数理解析研究所講究録
593巻 1986年 58-76

A Foundation of Analogical Reasoning: Analogical Union of Logic Programs

By

Makoto Haraguchi and Setsuo Arikawa

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

Abstract

This paper presents a theoretical foundation of analogical reasoning in which problem domains are represented by logic programs. Conceptually based on Gentner's structure mapping analogy, we formally define the analogical reasoning in terms of logic programming. Then, introducing the notion of analogical union, we characterize the reasoning in the deducibility from the analogical union. Based on this characterization, a general framework of analogical reasoning is also presented so that we can deal with various constraints for analogy.

1. Introduction

Based on some analogy between two or more domains in question, we often reason some unknown facts and knowledges to solve the problem concerning the domains. The reasoned facts are not necessarily true, however, the analogy under which the facts are reasoned gives an evidence showing that the fact may be true. Generally we must have rich enough knowledges of domains to perceive a suggestive analogy. Hence the power of analogical reasoning depends on the amount of knowledges about the domains and also depends on the power of deducing knowledges. From this viewpoint, we present in this paper a formalism of analogical reasoning in terms of deduction. Among possible systems of deduction, we consider in this paper a logic programming system based on Horn logic, since we now have a powerful computer system to perform it.

To give the formalism of analogical reasoning, the first problem is to precisely define the notion of analogy, even though the formalism cannot deal with our ambiguous and mysterious uses of analogy. Generally an analogy is a partial likeness between two or more domains, and the analogous domains can be regarded as the same one in some aspects. To view the different domains same, we must have a correspondence between the domains. Under this correspondence, some aspects of domains can be considered same. Hence, if we try to formalize the notion of analogy, we have to give the following:

(1) Representation language to describe the domains.

(2) An interpretation (semantics) of the representation.

(3) A definition of possible correspondences between the domains.

(4) A definition of which aspects of domains can be considered same under the possible correspondence.

Based on Polya[8], Gentner[3] and Winston[9,10], we define in this paper a formal analogy which is a relation of terms with the identification of facts. We assume that the domains are represented by logic programs which are finite sets of definite clauses, and also assume that the programs are interpreted as their least Herbrand models. Under this assumption, our correspondence between the domains is a relation of terms which satisfies the following axiom:

Axiom : for each n-ary function symbol f ($n \geqq 1$),

$$f(X_1, \ldots, X_n) \sim f(Y_1, \ldots, Y_n) \leftarrow X_1 \sim Y_1, \ldots, X_n \sim Y_n,$$

where $\sim$ is a predicate symbol to denote the relation of terms, and the terms denote elements in the domains. Then our formal analogy consists of the relation of terms and a set of paired facts (ground atoms) which are identical except for $\sim$. Thus the paired facts are compatible in the sense that they have the same predicate symbol. This definition is conceptually based on Polya's clarified analogy: "Two systems are analogous if they agree in clearly definable relations of their respective parts", where the systems and the parts correspond to logic programs and terms, respectively.

The second problem is to define the process of analogical reasoning. For this purpose, we consider in this paper the Winston's analogy-based reasoning[10], in which "similar" reasons are assumed to lead to "similar" effect. Introducing the notion of rule transformation, we formally define the analogy-based reasoning in terms of logic programming, and then precisely define the set of ground atoms reasoned by analogy.

Since we give the formalism of analogical reasoning in a system which performs deduction, the reasoning has some logical aspect. So the third problem is to characterize the reasoning in terms of deduction. For this purpose, we introduce the notion of analogical union of logic programs. The analogical union of programs P_1 and P_2 is itself a logic program which has copies of P_1 and P_2, and has some definite clauses concerning the predicate $\sim$. Then we show that ground atoms are reasoned by analogy iff they are logical consequences of the analogical union.

According to the previous studies of analogy, some semantic constraints are often required to the relation denoted by

$\sim$. Such semantic constraints can be expressed by some additional axioms for $\sim$. As an example of such a constraint, we consider the problem of analogy as a partial identity (Haraguchi and Arikawa [6]). Then the constraints for $\sim$ can be described as a first order theory. Also the analogy is actually a model for the theory. Based on this observation, we give a general framework of analogical reasoning so that it can deal with various constraints for analogy.

In Section 2, we briefly review the notion of structure-mapping analogy presented by Gentner[3]. In Section 3, we define our formal analogy and the reasoning based on it. In Section 4, we introduce the notion of analogical union, and characterize the reasoning in the deducibility from the analogical union. In Section 5, we give a general framework of analogical reasoning.

2. Structure-mapping analogy

Before we define the formal analogy, we briefly review the notion of structure-mapping analogy introduced by Gentner[3]. Her paper is concerned with the question of what makes some analogies useful in scientific thinking and others useless or harmful. To answer the question, she has considered the structure-mapping analogy between complex systems in order to characterize analogical models used in science, such as Rutherford's comparison of the atom to the solar system.

The structure-mapping analogy consists of a base system B (known domain), a target system T (domain in inquiry) with the following properties:

(P1) The target system T is described in terms of the base system B.

(P2) The objects in B are mapped on those in T, allowing the predicates of B to be applied in T.

Thus the structure-mapping analogy asserts that identical operations and relationships hold among non-identical objects. This assertion agree with the Polya's clarified analogy[8]: "Two systems are analogous if they agree in clearly definable relations of their respective parts."

Gentner has also assumed a "propositional network" of object nodes to represent knowledges in the domains. Given such a representation, she describes the structure-mapping analogy from the base system B to the target system T as follows:

(A1) There exists a mapping M of the nodes b_1, ..., b_n of B into the (different) nodes t_1, ..., t_n of T.

(A2) The mapping is such that substantial parts of the relational operational structure of B apply in T: that is, many of the relational predicates that are valid in B must also be valid in T, given the node substitutions dictated by M:

$$\mathrm{TRUE}[F(b_i,b_j)] \text{ implies } \mathrm{TRUE}[F(t_i,t_j)].$$

(A3) Relatively few of the valid attributes (the one-place predicates) within B apply validly in T:

$$\mathrm{TRUE}[A(b_i)] \text{ does not imply } \mathrm{TRUE}[A(t_i)].$$

It should be noticed that the assertion A3 is used to specify the relationship between B and T is one of analogical relatedness and not "literal similarity". Also the mapping M in A1 is required to be a one-to-one correspondence of object nodes.

On the other hand, Winston[9,10] has designed an analogical reasoning system in which each domain is internally represented by a network of frames. The analogy he has considered is a pairing of frames with the agreement of slot values of frames. Here the frames represent objects in the domains. The agreement of relation-slot values just coincides with the assertion A2. However the agreement of attribute-slot values may violate the assertion A3.

The purpose of the present paper is to formalize the analogical reasoning in terms of deduction as unrestrictedly as possible. Hence we do not require the assertion A3. Also we do not assume that the mapping M in A1 is one-to-one. Moreover both Gentner[3] and Winston [9,10] has considered the notion of higher-order relational predicates to characterize analogies. However they are extra-logical. Hence we consider in this paper only relational predicates among elements in the domains.

Now we state the outline of the formalism presented in this paper:

(O1) The source and the target systems are represented by logic programs, and are assumed to be the least models defined by the programs. The validity of relational predicates is logical in the sense that the predicates are logical consequences of the programs. Moreover each object (node) in the propositional network representation corresponds to a ground term.

(O2) The mapping M between the domains is defined by a pairing of ground terms, and is not necessarily one-to-one.

(O3) The identical relationships among non-identical objects are defined to be the syntactic identity except for the pairing in O2.

3. Reasoning based on formal analogy

we define in this section the formal analogy and the reasoning based on it. Since we assume that domains in question are represented by logic programs, we first give some necessary definitions concerning logic programs.

A definite clause is a clause of the form

$$A \leftarrow B_1, \ldots, B_n \quad (n \geqq 0),$$

where A and B_j are positive literals. We call the definite clause a rule. A logic program is a finite set of rules, and is simply called a program.

Since a program P is a set of clauses, any model for P can be considered as the corresponding Herbrand model. For instance, see [7]. Every Herbrand model for P has the same domain U(P), called the Herbrand universe, and the same meaning of function symbol appearing in P. U(P) is defined to be the set of all ground terms whose symbols appear in P. The meaning of n-ary function symbol f is defined to be a function

$$\lambda\ [[t_1,\ldots,t_n];f(t_1,\ldots,t_n)]\ :\ U(P)^n \rightarrow U(P).$$

We also need the notion of Herbrand base. The Herbrand base B(P) of a program P is defined as

$$B(P) = \{\ p(t_1,\ldots,t_n)\ \mid\ p \text{ is a n-ary predicate symbol appearing in P, and } t_j \in U(P)\}.$$

An element of B(P) is called an atom (ground atom).

Then, each Herbrand model(interpretation) is specified by a subset of B(P). According to the model intersection property[1], the intersection of all Herbrand models for P is also a model for P. This model is called the least model for P, and is denoted by M(P).

Proposition 3.1. ([1,7]) M(P) is the set of all ground atoms which are logical consequences of P.

According to Proposition 3.1, we take M(P) as the formal meaning of P, and call an element in M(P) a fact. In what follows, we consider only the Herbrand models, simply called models.

We define the correspondences of analogy by a pairing of elements in domains.

Definition 3.1. Let P_1 and P_2 be logic programs. A finite subset of $U(P_1) \times U(P_2)$ is called a pairing of terms. For a pairing ϕ, we define the set ϕ^+ to be the smallest set satisfying the following properties:

(3.1) $\phi \subseteq \phi^+$,

(3.2) if $\langle t_1, t'_1\rangle,\ldots,\langle t_n, t'_n\rangle \in \phi^+$, then $\langle f(t_1,\ldots,t_n), f(t'_1,\ldots,t'_n)\rangle \in \phi^+$, where f is a function symbol appearing in both P_1 and P_2.

As mentioned in the introduction, we must give the definition of which aspects of domains can be considered same under a pairing ϕ. We define it by a syntactic identity except for the pairing ϕ.

Definition 3.2. For a pairing ϕ, two ground atoms $\alpha \in B(P_1)$ and $\alpha' \in B(P_2)$ are said to be identified by ϕ, if (1)

α and α' are compatible, that is, they can be written as

$$\alpha = p(t_1, \ldots, t_n),$$
$$\alpha' = p(t'_1, \ldots, t'_n),$$

for some predicate symbol p, and (2) α and α' are syntactically same except for the pairing ϕ, that is, $\langle t_i, t'_i \rangle \in \phi^+$. This case is denoted by $\alpha \phi \alpha'$.

Since our domains under consideration are least models $M(P_1)$ and $M(P_2)$, ϕ defines a relation $ID(P_1, P_2; \phi)$ of facts as follows:

$$ID(P_1, P_2; \phi) =$$
$$\{ \langle \alpha, \alpha' \rangle \mid \alpha \in M(P_1),\ \alpha' \in M(P_2),\ \alpha \phi \alpha \}.$$

When we say that a pairing ϕ is an analogy, we generally require some constraints for ϕ. However, as a first step, we consider that the set $ID(P_1, P_2; \phi)$ represents the aspects of domains which can be viewed the same under the pairing ϕ. We give in Section 5 a general framework to cope with various constraints for analogy.

Now the analogical reasoning we consider is stated as follows:

Assume that, in P_1, the premises $\beta_1, \ldots, \beta_n$ logically imply a fact α. Also assume that the similar premises $\beta_1', \ldots, \beta_n'$ hold in P_2. Then we reason an atom α' similar to α.

It should be noticed that the reasoning stated in the above is conceptually due to Winston's analogy-based reasoning[10] based on the causal structures of domains. Since our similarity between $M(P_1)$ and $M(P_2)$ is the set $ID(P_1, P_2; \phi)$, we restate the statement above as follows:

Let $\beta_1, \ldots, \beta_n$ in $M(P_1)$ logically imply α in P_1. Moreover assume that there exist $\beta_1', \ldots, \beta_n'$ in $M(P_2)$ such that $\beta_j \phi \beta_j'$ for all j. Then we reason an atom α in $B(P_2)$ such that $\alpha \phi \alpha'$.

The reasoned atom α is not necessarily a logical consequence of P_2. Hence the reasoning go beyond a deduction. As mentioned in the introduction, our goal is to describe the reasoning in terms of deduction. For this purpose, we need the following definition:

Definition 3.3. Let

$$R_1 = (\alpha \leftarrow \beta_1, \ldots, \beta_n),$$
$$R_2 = (\alpha' \leftarrow \beta_1', \ldots, \beta_n')$$

be two ground rules ($n \geqq 1$) whose symbols are all appearing in P_1 and P_2, respectively. Let ϕ and I_j be a pairing and an Herbrand interpretation of P_j, respectively. Then the rules R_1 and R_2 are called ϕ-analogous w.r.t. I_1 and I_2, if $\beta_j \in I_1$, $\beta_j' \in I_2$, $\alpha \phi \alpha'$, and $\beta_j \phi \beta_j'$. In this case, R_1 (R_2) is called a ϕ-analogue of R_2 (R_1) w.r.t. ϕ.

We call the act of converting R_1 into R_2, or R_2 into R_1, a transformation of rules. In what follows, we represent the transformation by the following schema:

$$\frac{\alpha \leftarrow \beta_1, \ldots, \beta_n}{\alpha' \leftarrow \beta_1', \ldots, \beta_n'} \quad (\phi, I_1, I_2)$$,

where $\alpha \phi \alpha'$, $\beta_j \phi \beta_j'$, $\beta_j \in I_1$, $\beta_j' \in I_2$ and the dotted line shows that the upper rule is transformed into the lower rule. Using this schema, we can represent the reasoning we consider as follows:

$$\dfrac{\beta_1', \ldots, \beta_n' \qquad \dfrac{\dfrac{A \leftarrow B_1, \ldots, B_n}{\alpha \leftarrow \beta_1, \ldots, \beta_n}\ (\theta)}{\alpha' \leftarrow \beta_1', \ldots, \beta_n'}\ (\phi, M(P_1), M(P_2))}{\alpha'}$$,

where $A \leftarrow B_1, \ldots, B_n$ is a rule in P_1, θ is a ground substitution to obtain a logically true ground rule $\alpha \leftarrow \beta_1, \ldots, \beta_n$, and the last real line shows modus ponens. Thus the analogical reasoning is a combination of the usual deduction and the rule transformation. This schema is called fundamental.

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 example:

Example 3.1. Let P_1 and P_2 be the following programs:

$$P_1 = \{\ p(a,b) \leftarrow,\quad q(b) \leftarrow,\quad r(b) \leftarrow,\quad s(b) \leftarrow q(b),\ r(b)\},$$

$$P_2 = \{\ p(a',b') \leftarrow,\quad r(b') \leftarrow\ \}.$$

Then we have the following fundamental schema:

$$\frac{p(a',b') \qquad \dfrac{q(b) \leftarrow p(a,b)}{q(b') \leftarrow p(a',b')}\ (\phi, M(P_1), M(P_2))}{q(b')},$$

where $\phi = \{ \langle a,a'\rangle, \langle b,b'\rangle \}$. $q(b')$ is not a logical consequence of P_2. However we assume that we can make use of $q(b')$, as if it is a fact, to derive some new atoms further. According to this assumption, we can derive $s(b')$, since

$$\frac{q(b'),\ r(b') \qquad \dfrac{s(b) \leftarrow q(b),\ r(b)}{s(b') \leftarrow q(b'),\ r(b')}\ (\phi, M(P_1), M(P_2) \cup \{q(b')\})}{s(b')}$$

is a fundamental schema. Thus our assumption above allows a monotonic extension of models for P_2. We precisely define this extension.

Definition 3.4. For a given pairing ϕ, we define a set $M_i(*)$ for i=1,2 as follows:

$$M_i(*) = \cup_n M_i(n),$$

$$M_i(0) = M(P_i) = \{ \alpha \in B(P_i) \mid P_i \vdash \alpha \},$$

$$M_i(n+1) = \{ \alpha \in B(P_i) \mid R_i(n) \cup M_i(n) \cup P_i \vdash \alpha \},$$

where $S \vdash \gamma$ denotes that γ is a logical consequence of S, $R_i(n)$ is the set of all ground rules which are ϕ-analogues of ground instances of rules in P_j $(j \neq i)$ with respect to $M_j(n)$ and $M_i(n)$.

The following proposition asserts that our extension of least model $M(P_i)$ to $M_i(*)$ is admissible.

Proposition 3.2. For each i, $M_i(*)$ is a (Herbrand) model for P_i.

To prove this proposition, we need operators to give the fixpoint semantics of logic programs[1,7]. Let P and Pow(S) be a possibly infinite set of rules and the power set of a set S, respectively. Then we define an operator

$$T(P) : Pow(B(P)) \rightarrow Pow(B(P))$$

as follows: For a set $I \subseteq B(P)$, $\alpha \in T(P)(I)$ iff there exists a rule $A \leftarrow B_1,\ldots,B_n$ $(n \geq 0)$ in P and a ground substitution θ such that $A\theta = \alpha$ and $B_j\theta \in I$ for all j.

Proposition 3.3. ([1]) For a logic program P and a (Herbrand) interpretation I of P, the following conditions are equivalent:

(1) I is a model for P.

(2) $T(P)(I) \subseteq I$.

(3) $T(\{C\})(I) \subseteq I$ for any rule C in P.

Proof of Proposition 3.2. Let $A \leftarrow B_1,\ldots,B_n$ be a rule in P_i and θ be a ground substitution such that $B_j\theta \in M_i(*)$. Since $M_i(*) = \cup\ M_i(n)$, there exists a natural number N such that $B_j\theta \in M_i(N)$. Hence $M_i(N) \cup P_i \vdash A\theta$, and therefore

$$A\theta \in M_i(N+1) \subseteq M_i(*).$$

Hence $T(P_i)(M_i(*)) \subseteq M_i(*)$. Thus, according to Proposition 3.3, $M_i(*)$ is a model for P_i.

From this proposition, our extension $M_i(*)$ of $M(P_i)$ is admissible in the sense that $M_i(*)$ is a model for P_i. In the next section, we give a more logical characterization of $M_i(*)$.

4. Analogical union of logic programs

In this section, we introduce the notion of analogical union and study a logical aspect of analogical reasoning. Since we use the transformation of rules to derive atoms in $M_i(*)$, they are not necessarily logical consequences of P_i.

However the relation ϕ^+ of terms gives a correspondence between Herbrand universes, and the transformation is performed based on this correspondence. Hence we first program ϕ^+, and then program the transformation to characterize $M_i(*)$.

First we need a predicate symbol $\sim$ to denote the correspondence defined by ϕ^+, and also need the following rules:

For each pair $\langle t,t'\rangle$ in ϕ, $\quad t \sim t' \leftarrow .$

For each function symbol appearing in both P_1 and P_2,

$$f(X_1,\ldots,X_n) \sim f(Y_1,\ldots,Y_n) \leftarrow X_1 \sim Y_1,\ldots, X_n \sim Y_n.$$

The set of these rules is denoted by PAIR(ϕ). Then the following proposition is easily proved.

<u>Proposition 4.1</u>. For a pairing ϕ,

$$\langle t,t'\rangle \in \phi^+ \text{ iff } \mathrm{PAIR}(\phi) \vdash t \sim t' .$$

As Clark[2] has introduced the notion of completion of a logic program to justify the use of negation as failure rule, we introduce the notion of analogical union of two logic programs to justify the use of the transformation of rules. The analogical union of P_1 and P_2 is itself a logic program which has copies of P_i and has some additional definite clauses to perform the transformation. Each predicate symbol appearing in the copy of P_i has an index i to designate that the symbol comes from P_i. For simplicity, we replace each predicate symbol p in P_i with p_i.

The definite clauses to justify the transformation precisely describe the use of transformation to derive atoms. Formally, with each rule

$$p(t_1,\ldots t_n) \leftarrow \ldots, q(s_1,\ldots s_k),\ldots$$

in P_1, we associate the following clause:

$$\begin{aligned} p_2(W_1,\ldots,W_n) & \\ \leftarrow \quad & \ldots \\ & t_1 \sim W_1,\ldots, t_n \sim W_n, \\ & s_1 \sim V_1,\ldots, s_k \sim V_k, \\ & q_1(s_1,\ldots,s_k), \\ & q_2(V_1,\ldots,V_k), \\ & \ldots \quad , \end{aligned}$$

where W_i and V_j are introduced variables not appearing P_1 nor P_2. Similarly, with each rule

$$p(t_1,\ldots,t_n) \leftarrow \ \ldots,\ q(s_1,\ldots,s_k),\ \ldots$$

in P_2, we associate

$$\begin{aligned} p_1(W_1,\ldots,W_n) \leftarrow \quad & \ldots \\ & W_1 \sim t_1,\ldots,\ W_n \sim t_n, \\ & V_1 \sim s_1,\ldots,\ V_k \sim s_k, \\ & q_2(s_1,\ldots,s_k), \\ & q_1(V_1,\ldots,V_k),\ \ \ldots \end{aligned}$$

<u>Definition 4.1</u>. Let P_1 and P_2 be logic programs. Let ϕ be a pairing. The analogical union of P_1 and P_2, denoted by $P_1\phi\, P_2$, is the collection of definite clauses associated with each rule in P_i (i=1,2) together with PAIR(ϕ).

<u>Example 4.1</u>. Let

$$\begin{aligned} P_1 &= \{\ p(a,.b) \leftarrow, \\ &\qquad p(f(X),b) \leftarrow p(X,b)\ \}, \\ P_2 &= (p(f(c),d) \leftarrow\ \}, \\ \phi &= \{\ \langle a,f(c)\rangle,\ \langle b,d\rangle\ \}. \end{aligned}$$

Then the analogical union $P_1\phi\, P_2$ is

$$\begin{aligned} \{\ & p_1(a,b) \leftarrow, \\ & p_1(f(X),b) \leftarrow p_1(X,b), \\ & p_2(W_1,W_2) \leftarrow f(X) \sim W_1,\ b \sim W_2, \\ & \qquad\qquad X \sim V_1,\ b \sim V_2, \\ & \qquad\qquad p_1(X,b), \\ & \qquad\qquad p_2(V_1,V_2), \\ & p_2(f(c),d) \leftarrow, \\ & a \sim f(c) \leftarrow, \\ & b \sim d \leftarrow\ , \\ & f(X) \sim f(Y) \leftarrow X \sim Y\ \}. \end{aligned}$$

The reasoning defined in Section 3 is now characterized by the least Herbrand models for the analogical union.

<u>Theorem 4.1</u>. $M(P_1\phi\, P_2) = M'_1(*) \cup M'_2(*) \cup \phi\,'^{+}$, where

$$M'_i(*) = \{\ p_i(t_1,\ldots,t_n)\ \mid\ p(t_1,\ldots,t_n) \in M_i(*)\},$$
$$\phi\,'^{+} = \{\ s \sim t\ \mid\ \mathrm{PAIR}(\phi\ ) \vdash s \sim t\ \}\ .$$

In what follows, we rename each predicate symbol p in P_i

with p_i. Hence we can identify $M_i(*)$ with $M'_i(*)$. Also we identify ϕ^+ and ϕ'^+, according to Proposition 4.1.

Proof of Theorem 4.1. First we show

$$M(P_1 \phi P_2) \subseteq M_1(*) \cup M_2(*) \cup \phi^+.$$

Let $M(*)$ denote $M_1(*) \cup M_2(*) \cup \phi^+$. Since $M(P_1 \phi P_2)$ is the least model, it suffices to prove that $M(*)$ is a model for $P_1 \phi P_2$. In the proof below, we assume for the sake of simplicity that each rule in P_i is an assertion of facts or has a single literal in the body. Suppose that

$$\begin{aligned} C_0 : p_2(W_1,\ldots,W_n) \leftarrow\ & t_1 \sim W_1, \ldots, t_n \sim W_n, \\ & s_1 \sim V_1, \ldots, s_k \sim V_m, \\ & q_1(s_1,\ldots,s_m), \\ & q_2(V_1,\ldots,V_m) \end{aligned}$$

is in $P_1 \phi P_2$. From the definition of analogical union, the rule

$$C_1 : p_1(t_1,\ldots,t_n) \leftarrow q_1(s_1,\ldots,s_m)$$

should be in P_1. Also W_j and V_k never appear in C_1. Let θ be a ground substitution such that

$$\begin{aligned} & t_j\theta \sim W_j\theta \quad (\text{all } j), \\ & s_i\theta \sim V_i\theta \quad (\text{all } i), \\ & q_1(s_1\theta, \ldots, s_m\theta), \\ & q_2(V_1\theta, \ldots, V_m\theta) \end{aligned}$$

are in $M(*)$. Clearly $t_j\theta \sim W_j\theta$ and $s_i\theta \sim V_i\theta$ are in ϕ^+, and

$$\begin{aligned} & q_1(s_1\theta, \ldots, s_m\theta) \in M_1(*), \\ & q_2(V_1\theta, \ldots, V_m\theta) \in M_2(*). \end{aligned}$$

Hence there exists a natural number N such that

$$\begin{aligned} & q_1(s_1\theta, \ldots, s_m\theta) \in M_1(N), \\ & q_2(V_1\theta, \ldots, V_m\theta) \in M_2(N). \end{aligned}$$

Hence the rule

$$C_2 : p_2(W_1\theta, \ldots, W_n\theta) \leftarrow q_2(V_1\theta, \ldots, V_m\theta)$$

is a ϕ-analogue of

$$p_1(t_1\theta, \ldots, t_n\theta) \leftarrow q_1(s_1\theta, \ldots, s_m\theta),$$

which is an instance of rule C_1 in P_1, with respect to $M_1(N)$ and $M_2(N)$. Thus $C_2 \in R_2(N)$ (Definition 3.4) and therefore

$$R_2(N) \cup M_2(N) \vdash p(W_1\theta, \ldots, W_n\theta).$$

This implies that $p(W_1\theta, \ldots, W_n\theta) \in M_2(N+1) \subseteq M_2(*)$. Hence

$$T(\{C_0\})(M(*)) \subseteq M(*).$$

For a rule in $P_1 \phi P_2$ with $p_1(X_1, \ldots, X_k)$ as the head, we can give a completely similar proof. For a rule C in $P_i \subseteq P_1 \phi P_2$, Proposition 3.2 has already proved that $T(\{C\})(M(*)) \subseteq M(*)$. Hence, according to Proposition 3.3, $M(*)$ is a model for $P_1 \phi P_2$.

Conversely we must show that

$$M(*) = M_1(*) \cup M_2(*) \cup \phi^+ \subseteq M(P_1 \phi P_2).$$

According to Proposition 4.1 and the definition of $P_1 \phi P_2$, it suffices to prove

$$M_1(*) \cup M_2(*) = \cup_n (M_1(n) \cup M_2(n)) \subseteq M(P_1 \phi P_2).$$

We prove that $M_i(n) \subseteq M(P_1 \phi P_2)$ for i=1,2, by the induction on n. Suppose first that n=0. Then $M_i(0) = M(P_i)$. So the result is trivial. Next suppose that, for some $n > 0$, $M_i(n) \subseteq M(P_1 \phi P_2)$ holds for i=1,2. By the definition,

$$M_2(n+1) = M(P_2(n)),$$

where $P_2(n) = R_2(n) \cup M_2(n) \cup P_2$. Since $M(P_2(n))$ is the least model for $P_2(n)$, we show that

$$M_2(n+1) = M(P_2(n)) \subseteq M(P_1 \phi P_2)$$

by proving that $M(P_1 \phi P_2)$ is a model for $P_2(n)$. To prove this, it suffices to verify that, for each C in $P_2(n)$,

$$T(\{C\})(M(P_1 \phi P_2)) \subseteq M(P_1 \phi P_2).$$

Case 1: $C \in M_2(n) \subseteq P_2(n)$. By the induction hypothesis and the fact that C is a ground atom, we have

$$T(\{C\})(M(P_1 \phi P_2)) = \{C\} \subseteq M_2(n) \subseteq M(P_1 \phi P_2).$$

Case 2: $C \in P_2 \subseteq P_2(n)$. Since $P_2 \subseteq P_1 \phi P_2$, the result is trivial.

Case 3: $C = (\alpha' \leftarrow \beta'_1, \ldots, \beta'_k) \in R_2(n) \subseteq P_2(n)$. By the definition of $R_2(n)$, there exists a rule

$$C_1 : A \leftarrow B_1, \ldots, B_k$$

in P_1 and a ground substitution θ such that

$$B_1\theta, \ldots, B_k\theta \in M_1(n),$$
$$\beta'_1, \ldots, \beta'_k \in M_2(n),$$
$$A\theta \sim \alpha', \; B_j\theta \sim \beta'_j \text{ (all j)}.$$

Since $M_2(n) \subseteq M(P_1 \phi P_2)$, $T(\{C\})(M(P_1 \phi P_2)) = \{\alpha'\}$. For the sake of simplicity, we assume that k=1, and write C and C_1

as follows:

$$C_1 \; : \; p_1(s) \leftarrow q_1(t),$$
$$C \; : \; p_2(s') \leftarrow q_2(t'),$$

where $\alpha'=p(s')$, $s\theta \sim s'$, $t\theta \sim t'$, $q_1(t\theta) \in M_1(n)$ and $q_2(t')$ $\in M_2(n)$. From the definition, $P_1 \phi P_2$ has

$$C_2 \; : \; p_2(W) \leftarrow s \sim W,\ t \sim V,\ q_1(t),\ q_2(V) \quad ,$$

where the variables W and V never appear in s nor t. Let

$$\sigma = \theta \cup \{ W \leftarrow s',\ V \leftarrow t' \}.$$

Then, by the induction hypothesis, we have

$$q_1(t\sigma) = q_1(t\theta) \in M_1(n) \subseteq M(P_1 \phi P_2),$$
$$q_2(V\sigma) = q_2(t') \in M_2(n) \subseteq M(P_1 \phi P_2).$$

Hence, by the definition of $T(\{C_2\})$, we have

$$\alpha' = p_2(s') = p_2(W\sigma) \in T(\{C_2\})(M(P_1 \phi P_2)).$$

Since $C_2 \in P_1 \phi P_2$, this implies that $\alpha' \in M(P_1 \phi P_2)$.

As a result, we have proved that $M_2(n+1) \subseteq M(P_1 \phi P_2)$. $M_1(n+1) \subseteq M(P_1 \phi P_2)$ is similarly proved. Hence we have $M_1(n) \cup M_2(n) \subseteq M(P_1 \phi P_2)$ for all n. This completes the proof.

5. General Framework of analogical reasoning

Based on the results of previous sections, we present in this section a general framework of analogical reasoning. Definition 3.4 formalizes the process of analogical reasoning, given the underlying pairing ϕ. Also Theorem 4.1 logically characterizes the reasoning in the deducibility from the analogical union.

Hence, once some pairing is "detected", the problem of analogy can be solved in the framework of deduction. On the other hand, it is leaved as an important problem to describe the notion of analogy detection. Here the problem of analogy detection is to find a pairing which satisfies certain constraints. As discussed before, the only constraint required so far is

$$A_0 : f(X_1, \ldots, X_n) \sim f(Y_1, \ldots, Y_n) \leftarrow X_1 \sim Y_1, \ldots, X_n \sim Y_n.$$

As mentioned in Section 2, Gentner[3] has required that a possible pairing of analogy is one-to-one.

Definition 5.1. ([6]) For programs P_1 and P_2, a pairing ϕ is called a partial identity if ϕ^+ is a one-to-one relation of terms.

If we require that ϕ is a partial identity, we need the following new axioms:

$$A_1 : X =_1 Y \leftarrow X \sim Z, Y \sim Z,$$
$$A_2 : X =_2 Y \leftarrow Z \sim X, Z \sim Y,$$

where $=_i$ is a predicate symbol to denote the identity relation on each Herbrand universe. Due to Clark[2], the following theory, denoted by EQ_i, is sufficient for $=_i$ to denote the identity relation:

1. $c \neq_i d$, for all pairs of distinct constants c, d in P_i.
2. $f(X_1, \ldots, X_n) \neq_i g(Y_1, \ldots, Y_m)$,
   for all pairs of distinct function symbols f, g in P_i.
3. $f(X_1, \ldots, X_n) \neq_i c$,
   for all pairs of constant c and function f in P_i.
4. $t[X] \neq_i X$, for each non-variable term $t[X]$ in P_i.
5. $X_j =_i Y_j \leftarrow f(X_1, \ldots, X_n) =_i f(Y_1, \ldots, Y_n)$,
   for each function f in P_i.
6. $X =_i X$.
7. $f(X_1, \ldots, X_n) =_i f(Y_1, \ldots, Y_n) \leftarrow X_1 =_i Y_1, \ldots, X_n =_i Y_n$,
   for each function f in P_i.
8. $p(Y_1, \ldots, Y_n) \leftarrow X_1 =_i Y_1, \ldots, X_n =_i Y_n, p(X_1, \ldots, X_n)$,
   for each predicate p in P_i.

Then the constraint for ϕ to be a partial identity is written as the following theory $CT(\phi)$:

$$CT(\phi) = EQ_1 \cup EQ_2 \cup \{A_0, A_1, A_2\} \cup \phi.$$

Then it is clear that ϕ is a partial identity iff $CT(\phi)$ is consistent. It should be noticed that a model for $CT(\phi)$ can define a partial identity. In other words, the analogy we desire is a model for the theory $CT(\phi)$. Based on this fact, we now give a general framework of analogical reasoning.

Problem of analogy detection: Given P_1 and P_2, find a pairing ϕ such that CT(ϕ) is consistent.

Once the problem above is solved, we proceed to the next step, due to Theorem 4.1:

Problem of reasoning based on the pairing: Given P_1, P_2 and ϕ , deduce some "useful" information from the analogical union $P_1 \phi P_2$ with respect to ϕ .

The authors[6] has already given a effective solution to the problem of analogy detection, provided that the corresponding pairing is a partial identity. Even when we make CT(ϕ) to be an another constraint rather than the partial identity, our framework based on the analogical union still stands to describe analogical reasoning.

6. Concluding remarks

We have presented in this paper a formalism of analogical reasoning in terms of logic programming based on Horn logic. Since we cannot deal with negative literals in the definite clauses, we have not paid attention to the uses of negative informations. However we can make use of them to reject some "wrong" analogies. In fact, the negative informations are described as formulas, and added to the theory CT(ϕ) in order to make ϕ not define a wrong analogy. Also the search space of possible pairings is reduced if we consider such a theory. From this viewpoint, a theory of analogical reasoning which deals with negative informations is under developing.

References

[1] Apt, K.R. and van Emden, M.H. (1982): Contribution to the theory of logic programming, JACM, 29, 3, 841-862.

[2] Clark, K.L. (1978): Negation as Failure, in Logic and Databases, H. Gallaire and J. Minker (Eds.), Plenum Press, New York, 293-322.

[3] Gentner, P. (1982): Are scientific analogies metaphors?, in Metaphor: Problems and Perspectives, D.S. Miall (Ed.), The Harvester Press, Sussex, 106-132.

[4] Haraguchi, M. and Arikawa, S. (1985): Analogical reasoning based on the theory of analogy, Res. Rept. Inst. Fund. Inform. Sci. Kyushu Univ., No. 105.

[5] Haraguchi, M. (in press): Analogical reasoning using transformations of rules, Bull. of Infor. Cybernetics, 22.

[6] Haraguchi, M. and Arikawa, S. (in prep.): Analogical reasoning based on the partial identity between least Herbrand models.

[7] Lloyd, J.W. (1984): Foundations of logic programming, Springer-Verlag.

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

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

[10] Winston, P.H. (1983): Learning new principles from precedents and exercises, Artificial Intelligence, 19, 321-350