

Partially Isomorphic Generalization and Analogical Reasoning*

Eizyu Hirowatari ** and Setsuo Arikawa

Research Institute of Fundamental Information Science,
Kyushu University 33, Fukuoka 812, Japan
e-mail: {eizyu, arikawa}@rifis.sci.kyushu-u.ac.jp

Abstract. Analogical reasoning is carried out based on an analogy which gives a similarity between a base domain and a target domain. Thus the analogy plays an important role in analogical reasoning. However, computing such an analogy leads to a combinatorial explosion. This paper introduces a notion of partially isomorphic generalizations of atoms and rules which makes it possible to carry out analogical reasoning without computing the analogy, and gives a relationship between our generalization and the analogy. Then we give a procedure which produces such a generalization in polynomial time.

1 Introduction

Analogical reasoning is an important paradigm of machine learning. It acquires unknown knowledge by computing an analogy which gives a similarity between a base domain and a target domain. In analogical reasoning, we first detect an analogy, and then project the well-known knowledge in the base domain into the target domain under the analogy. Thus essentials of analogical reasoning are computing an analogy which is a mapping from a base domain to a target domain. However, there often arises a problem of combinatorial explosion in computing analogies. We solve this problem by using a new concept of partially isomorphic generalizations of atoms or rules.

Our partially isomorphic generalization is a method to generalize an atom or a rule as general as possible without destroying its syntactical structure. We show that the facts, i.e., grand atoms, derived from the atoms thus generalized are also derived by the ordinary analogical reasoning by Haraguchi and Arikawa [1, 2]. Hence our generalization can be justified by their theory of analogical reasoning.

2 Analogical Reasoning

Analogical reasoning is carried out by projecting some of a base domain to a target domain under an analogy. Hence we take an analogy as a mapping from the base to the target.

* This work is partly supported by Grants-in-Aid for JSPS fellows and Scientific Research on Priority Areas from the Ministry of Education, Science and Culture, Japan.

** JSPS Fellowship for Japanese Junior Scientists.

In this paper we deal with logic programs (programs, for short) as the domains for analogical reasoning. Let P_1 and P_2 be base and target programs, respectively, on which analogical reasoning is carried out. Let U_i be the Herbrand universe for P_i ($i = 1, 2$). For a finite subset φ of $U_1 \times U_2$, we define a set φ^+ as the smallest set that satisfies the following conditions:

- (a) $\varphi \subseteq \varphi^+$,
- (b) $\langle t_1, s_1 \rangle, \dots, \langle t_n, s_n \rangle \in \varphi^+ \Rightarrow \langle f(t_1, \dots, t_n), f(s_1, \dots, s_n) \rangle \in \varphi^+$,

where f is a function symbol occurring in both P_1 and P_2 . We say that φ is an *analogy* if we can take φ as a function such that $\varphi^+(t) = s$ if $\langle t, s \rangle \in \varphi^+$.

An analogy can be extended from terms to atoms in a natural way. For atoms $\alpha = p(t_1, \dots, t_n)$ and $\beta = p(s_1, \dots, s_n)$, and for an analogy φ , we say that α is *analogous to* β *under* φ , if $\varphi^+(t_i) = s_i$ holds for $1 \leq i \leq n$.

3 Partially Isomorphic Generalization

For atoms α and β , we write $\alpha \leq \beta$ when $\beta\theta = \alpha$ for a substitution θ , and $\alpha \simeq \beta$ when $\beta \leq \alpha$ and $\alpha \leq \beta$. Let S be the set of all generalizations of an atom. By $[S]$ we denote the set of equivalence classes of all atoms in S induced by \simeq . From now on we identify an atom with its equivalence class. Then \leq is a partial order on $[S]$. We define two binary functions \sqcap and \sqcup on $[S]$ as follows: For $[\alpha]$ and $[\beta]$ in $[S]$, $[\alpha] \sqcap [\beta]$ and $[\alpha] \sqcup [\beta]$ are the greatest common instance and the least common generalization of $\{[\alpha], [\beta]\}$ w.r.t. \leq , respectively [5]. Hence $[S]$ is a lattice with a partial order \leq , a meet operator \sqcap , and a join operator \sqcup . We call the lattice $[S]$ a *normal lattice*.

Now we introduce a new concept of partially isomorphic generalizations of atoms. Let α be an atom and t be a term occurring in α . Then we say that t is a *quasi-replaceable term* of α , if t is a constant or a term of the form $f(X_1, \dots, X_n)$, where f is a function symbol and each X_i is a variable symbol. For u , a term or an atom, let V_u be the set of all variables occurring in u . Let $\alpha[t]$ be the atom obtained by replacing each occurrence of a quasi-replaceable term t in α by a new variable Z . If $V_t \cap V_{\alpha[t]} = \emptyset$ holds, we write $\alpha \rightarrow \beta$ for each variant β of $\alpha[t]$, and then we say that t is a *replaceable term* of α . The relation \rightarrow is a binary relation on a set of atoms. We define \rightarrow^* as the reflexive and transitive closure of \rightarrow . Then we say that β is a *partially isomorphic generalization* (PIG, for short) of α , if $\alpha \rightarrow^* \beta$.

Let S be the set of all PIGs of an atom. We consider the set $[S]$ of equivalence classes of all atoms in S induced by \simeq . Just as we have done with \leq , we have the following theorem.

Theorem 1. *Let S be the set of all PIGs of an atom. Then $[S]$ is a lattice with a partial order \rightarrow^* , a meet operator \sqcap , and a join operator \sqcup .*

We call the lattice $[S]$ a *PIG lattice*. There exists an atom γ such that $\beta \rightarrow^* \gamma$ holds for each PIG β of α . We say such an atom γ to be the *greatest PIG* of α . The difference of the PIG lattice from the normal lattice is shown in Figure 1.

Let α be a ground atom and φ be an analogy. Then we define $Ana(\alpha, \varphi)$ as the set of all ground atoms to which α is analogous under φ , and $G(\alpha)$ as the

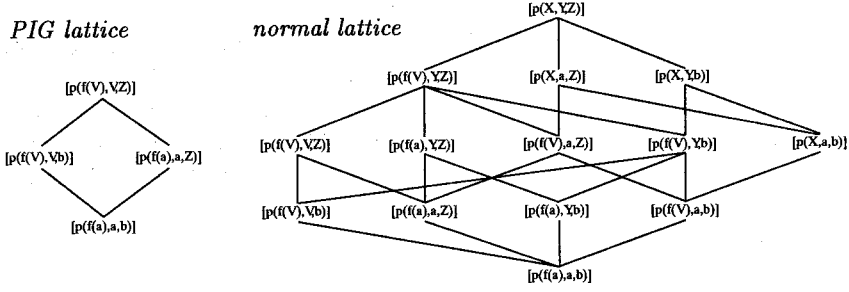


Fig. 1. The PIG lattice and the normal lattice for $p(f(a), a, b)$.

set of all ground instances of the greatest PIG of α . Then we have the following theorem.

Theorem 2. *Let α be a ground atom. For each atom β in $G(\alpha)$, there exists an analogy φ such that β is in $Ana(\alpha, \varphi)$.*

A ground atom α is analogous to all ground instances of the greatest PIG of α under analogies. Hence, in order to obtain an analogy φ and a grand atom to which α is analogous under φ , it suffices to compute the greatest PIG of α . Now we show an algorithm which computes the greatest PIG of an atom.

Algorithm: G_{PIG}

input: an atom α

output: the greatest PIG A of α

begin

$A := \alpha$; $N := 0$;

while there exists the $(N + 1)$ -st quasi-replaceable term t from the right in A **do**

if t is a replaceable term **then**

 replace each occurrence of t in A by a new variable Z

else $N := N + 1$;

 output A and halt

end.

The *length* of an atom α is the number of occurrences of constant, variable and function symbols in α . Then the numbers of nodes in the PIG lattice and the normal lattice for the atom of length n is at most 2^n and $en!$, respectively, where e is Napier's number.

The following theorem guarantees that we can compute the greatest PIG of an atom in polynomial time.

Theorem 3. *Let α be an atom of length n . The greatest PIG of α can be computed in time $O(n^2)$.*

We have realized the PIG system as a Prolog program which takes an atom α as input, constructs PIGs of α in an ordering by the relation \rightarrow , and then returns the greatest PIG of α as output. It works based on the algorithm G_{PIG} . The system has been implemented by K-Prolog on Spark Station 10. For example, if the question to our PIG system is ‘‘?- pig(p(f(a),a,g(b),g(b)),Atom).’’, then the answer from the system is ‘‘Atom = p(f(_222),_222,_218,_218)’’.

4 Reasoning by PIG

Just as we have done with PIGs of atoms, we can define PIGs of rules with bodies, and we have the same results on rules as those of atoms.

Now we discuss reasoning by PIGs. Let P_1 and P_2 be programs. For each rule C in $P_1 \cup P_2$, we compute the greatest PIG R of C in polynomial time, and then learn a new program P obtained by replacing each C in $P_1 \cup P_2$ by R . Thus we can acquire the fact derived from P without computing an analogy which often leads to a combinatorial explosion. The fact thus acquired can be derived from P_1 and P_2 by analogical reasoning. Hence reasoning by PIGs of rules is more useful than the analogical reasoning as far as time complexity is concerned.

5 Conclusion

The ordinary methods of generalization [4] of examples often cause non-valid and over generalization, and sometimes they need vast search-spaces. To overcome these difficulties we have considered syntactic analogies and introduced the notion of PIG. Our PIGs are all valid generalizations in the sense that they are justified by the theory of analogical reasoning. Moreover each PIG can be computed in polynomial time.

We are now considering a declarative definition of PIGs, and a kind of completeness of PIGs with respect to the analogical reasoning. Also we are improving our previous work on EBG by analogical reasoning [3] using PIGs.

References

1. Haraguchi, M., Arikawa, S.: A foundation of reasoning by analogy – analogical union of logic programs. In Proceedings of Logic Programming Conference (1986), Lecture Notes in Computer Science **264** (1987) 58–69
2. Haraguchi, M., Arikawa, S.: Reasoning by analogy as a partial identity between models. In Proceedings of First International Conference on Analogical and Inductive Inference, Lecture Notes in Computer Science **265** (1987) 61–87
3. Hirowatari, E., Arikawa, S.: Explanation-based generalization by analogical reasoning. In Proceedings of International Workshop on Inductive Logic Programming (1992)
4. Plotkin, G. D.: A note on inductive generalization. Machine Intelligence **5** (1970) 153–216
5. Reynolds, J. C.: Transformational systems and the algebraic structure of atomic formulas. Machine Intelligence **5** (1970) 135–151