ICOT Technical Memorandum: TM-0476

TM-0476

Learning General Rules with Exceptions: An Application of Circumscription

by
J. Arima

March, 1988

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Learning General Rules with Exceptions: An Application of Circumscription

Jun Arima

ICOT Research Center
Institute for New Generation Computer Technology
Mita Kokusai Bldg. 21F
4-28 Mita 1-chome, Minato-ku, Tokyo, 108, Japan
Phone: +81 3 456 4365, C.Mail Address: arima%icot.jp@relay.cs.net

Abstract

This paper attempts to formalize a general and real type of learning, learning general rules with exceptions. For this purpose, the circumscription technique is used, and a new form is proposed. The possibility that our new form is collapsed into first order logic is also discussed.

Paper length: 4000 words.

Topic: Knowledge Representation.

Keywords: Formula Circumscription.

1. Introduction

Research of non-monotonic reasoning is an attempt to formalize aspects of non-monotonicity in human reasoning processes. Many researchers seem to have concentrated their efforts on narrow applications. This paper proposes a new application of non-monotonic reasoning, learning general rules with exceptions and attempts to formalize it using formula circumscription [4]. It also proposes a new form, called *directional generalization rule*, and discusses the possibility that our new form is collapsed into first order logic.

McCarthy provides circumscription [3, 4] as a form of non-monotonic reasoning. Formula circumscription is one version of circumscription. It is a general formulation in that a wff is minimized, whereas the earlier form minimizes some predicates. Formula circumscription is a powerful formulation to express non-monotonic reasoning, but in order to utilize it, we must specify many predicate parameters. Especially, the wff to be minimized seems the most difficult but the most important to be given, because there are no constraints to decide it and the problem is left entirely in our hands. Moreover, to give the wff means to select one of some types of reasoning by giving predicates the intended interpretation. Hence, how to specify the wff is a problem. In this paper, limiting the use of formula circumscription in an application, it is used and specified in a generalization rule, and it is shown that formula circumscription is a useful approach to formalize some aspects of learning.

The second section introduces a specialized form of formula circumscription, which is simple and will be helpful to understand not only our arguments after that but also what formula circumscription can express. It is called *partially directional circumscription*. Its intuitive idea is to assign the same value to entities with respect to a certain property, P, as a certain property, Λ, as far as possible in the domain which satisfies a certain property, Φ. Namely, its formulation means that entities which satisfy property P generally satisfy property Λ among the entities which satisfy property Φ. In this sense, we say that P *is directional to* Λ *inside* Φ or P *is directed to* Λ *inside* Φ.

In the third section, partially directional circumscription is applied to formalization of a particular aspect of learning, learning general rules with exceptions, and a new form, *directional generalization rule*, is proposed. There, a comparative binary relation on the number of elements satisfying each property which occurs as its arguments is introduced. Depending on the relations, we select the directivity of P. Intuitively, the form expresses the idea that, inside a certain domain Φ, most entities have a certain property P if there are much more entities satisfying the property P than not so far as we know.

The fourth section explores how to use our new formulation and the possibility that it is collapsed into first order logic.

2. Partially Directional Circumscription: A Special Formulation of Formula Circumscription

To obtain a special and important class of formula circumscription, consider the simplest version, predicate circumscription, and consider what character the formulation has. In learning processes, some inductive reasoning is necessary. However, predicate circumscription is clearly not sufficient for formalizing such inductive reasoning directly, because simply minimizing the extension of some properties cannot formalize induction which generalizes such properties. For this reason, we try to extend predicate circumscription. Predicate circumscription can make the extension of property P minimal. When we consider that it makes the extension of P closest to the extension of 'false' (its extension means empty set and it is denoted by $\perp$) as far as possible, we can extend predicate circumscription.

That is, what we want is a formulation which can give an extension of P so that the extension will be closest to the extensions of our intended predicates. With such formulation, it will be possible to give P the intended meaning and to realize flexible reasoning, not only induction. Moreover, even if new information contradicts the old result of formulation, the result is revised non-monotonically because such formulation

does not fix the extension of P to our intended extension, but simply makes the extension closest to ours.

This is the basic idea of the formulation which is proposed in this section. However, some refinement is still necessary. In many applications, it is necessary to change the interpretation of P according to the properties of entities. Therefore, we propose a formulation so that we can change the *directivity* of P, that is, the extension which P is intended to be closest to, according to property Φ, and the formulation is called "*partially directional circumscription*".

In this paper we write t instead of a tuple of finite terms for brevity. By n-ary predicate, we mean an expression, $\lambda x.(\alpha(x))$, where x is a tuple of n variables and $\alpha(x)$ is a formula in which x occurs free and no other variables occur free. That is, a predicate is obtained from a formula by λ-abstracting all of the free variables in it.

Let P be a tuple of distinct predicate symbols, $P1,\cdots,Pn$, and Ψ a tuple of predicates, $\Psi 1,\cdots,\Psi n$, where Pi and Ψi have the same arity. $[\Psi/P]$ means a substitution, representing $[\Psi 1/P1,\cdots,\Psi n/Pn]$ and usually abbreviated $[\Psi]$. We write $\alpha(x)[\Psi/P]$ for the result of replacing simultaneously each occurrence Pi in $\alpha(x)$ by Ψi. And $\forall x.(P(x) \equiv \Psi(x))$ means $\forall x.(P(x) \supset \Psi(x)) \wedge \forall x.(P(x) \subset \Psi(x)))$.

Definition [*partially directional circumscription*].
Let P be a tuple of distinct predicate symbols, and let A be a formula. The *partially directional circumscription of* P *to* Λ *inside* Φ is

$$A[P] \wedge \forall \rho.(A[\rho] \wedge \forall x.((\Phi(x) \supset ((P(x) \equiv \Lambda(x)) \supset (\rho(x) \equiv \Lambda(x)))) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \rho(x))))), \quad \text{(D1)}$$

,where Φ and Λ are predicates in which no predicate symbols in P occur. This formula is denoted by Pd-circum(A; P ~ Λ / Φ), and asserts that the extension of P cannot be made closer to the extension of Λ inside an extension of Φ. That is, intuitively, it assigns the

same truth value to each entity satisfying Φ with respect to P as with respect to Λ as far as A holds.

In this paper, $\perp$ represents a property, $\lambda x.(\text{false})$, with respect to which for all tuples of entities the false value is assigned constantly, and similarly, $\top$ represents a property, $\lambda x.(\text{true})$, with respect to which the true value is assigned constantly.

Proposition 1.

1) Pd-circum(A; P ~ $\perp$ / $\top$) $\equiv$ Circum(A; P), where Circum(A; P) is the predicate circumscription of P in A.

2) Partially directional circumscription is a special formulation of formula circumscription.

Proofs.

1) By predicate calculus.

2) Partially directional circumscription is equivalent to the formula circumscription which minimizes the wff $\neg(\Phi(x) \supset (P(x) \equiv \Lambda(x)))$.

Proposition 1 says that predicate circumscription is to make some predicates directional to false in a whole domain.

3. Directional Generalization Rule:
a Form of Learning General Rules with Exceptions

One aim of concept learning [5], it can be considered to be to obtain some properties which, for some set of entities, all entities have in common. Here, more generally, we want to consider a certain type of learning, in which one aim is to obtain some properties (or knowledge) which, for some set of entities, most entities have in common. One of the obvious reasons why such learning general rules is important is that every rule generally has its exceptions in itself.

After this, we consider a case that P is directional to either $\bot$ or $\top$ inside Φ, not only because it makes our problem simpler, but because it seems to be enough useful to formalize such learning at present. We must still choose negative or positive directivity of the property inside each domain. On what should the judgment depend? The key idea of this solution is very natural. We introduce a comparative relation on the number of elements, and, depending on it, we select the majority. We represent it as '$>>$'. '$>>$' is a binary relation and its parameters are predicates. We say that α *surpasses* β and write '$\alpha >> \beta$'. Its intended meaning is that the number of entities satisfying property α is so much greater than the other that we can consider unknown entities to satisfy property α rather than β.

For all predicate α, β and γ:

Axioms on '$>>$'

I $(\alpha >> \beta) \wedge (\beta >> \gamma) \supset (\alpha >> \gamma)$ (A1.1)

II $\neg(\alpha >> \alpha)$ (A1.2)

III $(\alpha >> \beta) \wedge \forall x.(\alpha(x) \supset \gamma(x)) \supset (\gamma >> \beta)$ (A1.3)

IV $(\alpha >> \beta) \wedge \forall x.(\gamma(x) \supset \beta(x)) \supset (\alpha >> \gamma)$ (A1.4)

Next two axioms is not essential, but here, we consider these as axioms.

V a. $\alpha \neq \bot \supset (\alpha >> \bot)$ (A2.1)

VI b. $\alpha \neq \top \supset (\top >> \alpha)$, (A2.2)

where $\alpha = \beta$ denotes $\forall x.(\alpha(x) \equiv \beta(x))$ and $\alpha \neq \beta$ denotes $\neg\forall x.(\alpha(x) \equiv \beta(x))$.

Surpassing relation '$>>$' may be given by using a function to count up entities which satisfy some conditions and by using some adequate evaluate functions for '$>>$', or may be given directly based on our intuitions.

Now we introduce a formulation on learning general rules. Its idea is that, in a certain domain K, most entities have a certain property P if there are much more entities satisfying the property than not so far as we know. We want to consider that the entities which is shown to satisfy a certain property from facts are all that we know satisfy the property. For this purpose, we use the following two predicates.

For some properties $\ulcorner P \urcorner$ and $\llcorner P \lrcorner$, let

$$A[\llcorner P \lrcorner] \wedge \forall x.(\llcorner P \lrcorner(x) \supset P(x)) \qquad \text{(M1)}$$

and

$$A[\ulcorner P \urcorner] \wedge \forall x.(P(x) \supset \ulcorner P \urcorner(x)) \qquad \text{(M2)}$$

hold.

(M1) expresses that an extension of $\llcorner P \lrcorner$ that is smaller than or equal to an extension of P, where $\llcorner P \lrcorner$ represents such an extension of P. That is, $\llcorner P \lrcorner$ expresses a minimal extension of P, and similarly, $\ulcorner P \urcorner$ expresses a maximal extension of P. Using these predicates, the formulation which we want is partly as follows.

$$\lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x)) >> \lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x)) \supset \text{Pd-circum}(A ; P \sim \top / \Phi) \qquad \text{(L1)}$$

$\lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x))$ expresses a minimal set of entities which exist inside Φ and satisfy P. For $\forall x.(\neg \ulcorner P \urcorner(x) \supset \neg P(x))$, $\lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x))$ expresses a minimal set of entities which exist inside Φ and do not satisfy P. That is, (L1) declares that under A, P should be directional to $\top$ inside Φ if there are more entities satisfying P than not inside Φ so far as we know. Of course, it means that most entities satisfying Φ have a property P and, hence, anything that has a property Φ should be considered to have the property P if there is nothing in knowledge A that prevents it from doing so.

Here, we use a simpler form of (L1). If we substitute $\ulcorner P \urcorner$ for ρ in Pd-circum(A ; P $\sim$ $\top$ / Φ), in the result Pd-circum(A ; P $\sim$ $\top$ / Φ) is

$$A[P]$$

$$\wedge (A[\ulcorner P \urcorner] \wedge \forall x.(\Phi(x) \supset (P(x) \supset \ulcorner P \urcorner(x))) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \ulcorner P \urcorner(x)))). \qquad \text{(L2)}$$

A[P] and the antecedent of remaining part of (L2) follows from A and (M2). Consequently, we can obtain a simpler form of (L1),

$$\lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x)) >> \lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x)) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \ulcorner P \urcorner(x))). \qquad \text{(L3)}$$

With respect to '$\lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x) >> \lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x))$', we can similarly obtain

$$\lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x) >> \lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x)) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \llcorner P \lrcorner(x)) \qquad \text{(L4)}$$

Now from A, (M1), (M2), (L3) and (L4) we can get a form of learning general rules with exceptions,

$$\begin{aligned}
&A[P] \\
&\quad \wedge \forall \llcorner P \lrcorner, \ulcorner P \urcorner.(\\
&\qquad A[\llcorner P \lrcorner] \wedge \forall x.(\llcorner P \lrcorner(x) \supset P(x)) \\
&\qquad \wedge A[\ulcorner P \urcorner] \wedge \forall x.(P(x) \supset \ulcorner P \urcorner(x)) \\
&\qquad \supset (\lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x)) >> \lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x)) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \ulcorner P \urcorner(x)) \\
&\qquad \wedge (\lambda x.(\Phi(x) \wedge \neg \ulcorner P \urcorner(x) >> \lambda x.(\Phi(x) \wedge \llcorner P \lrcorner(x)) \supset \forall x.(\Phi(x) \supset (P(x) \equiv \llcorner P \lrcorner(x)) \\
&\qquad\qquad). \qquad \text{(D2)}
\end{aligned}$$

We refer to this form as the directional generalization rule and this is denoted by Directional-G(A; P / Φ).

Example 1:

In a DB with (F1), three birds, tweety, jack and p-suke are registered, and the information, "p-suke cannot fly" is given. This may be represented as follows:

$$\text{Bird(tweety)} \wedge \text{Bird(jack)} \wedge \text{Bird(p-suke)} \wedge \neg \text{Fly(p-suke)}. \qquad \text{(E1.1)}$$

Also, assume that enough information including (A1.1~4) and (A2.1~2) on surpassing relation '>>' is given. That is, assume the following knowledge except for the above basic axioms:

$$\lambda x.(x=\text{tweety}) << \lambda x.(x=\text{jack} \vee x=\text{p-suke})$$
$$\wedge\, \lambda x.(x=\text{jack}) << \lambda x.(x=\text{p-suke} \vee x=\text{tweety})$$
$$\wedge\, \lambda x.(x=\text{p-suke}) << \lambda x.(x=\text{tweety} \vee x=\text{jack}). \qquad \text{(E1.2)}$$

Now consider this with respect to Fly. In this case, $\perp$ clearly satisfies the condition of $\llcorner \text{Fly} \lrcorner$ as $\llcorner P \lrcorner$ in (M1), and $\lambda x.(\neg x=\text{p-suke})$ for $\ulcorner \text{Fly} \urcorner$†. Therefore, using (E1.1),

$$\lambda x.(\,\text{Bird}(x) \wedge \llcorner \text{Fly} \lrcorner(x)) = \perp, \qquad \text{(E1.3)}$$
$$\lambda x.(\,\text{Bird}(x) \wedge \neg \ulcorner \text{Fly} \urcorner(x)) = \lambda x.(x=\text{p-suke}). \qquad \text{(E1.4)}$$

Now from (A2.1)

$$\lambda x.(x=\text{p-suke}) >> \perp \qquad \text{(E1.5)}$$

follows. So, using (E1.3)~(E1.4) and Directional-G(A; Fly / Bird), it gives

$$\forall x.(\,\text{Bird}(x) \supset \neg \text{Fly}(x)\,). \qquad \text{(E1.6)}$$

---

† How to compute these generally is beyond the scope of this paper. However, Lifschitz proposes a useful way in the case that a givenformula satisfies a certain class[2]. And next section will show a solution based on the Lifschitz's way.

This result shows generalization of knowledge. From (E1.4) and (E1.1) we can see that both Tweety and Jack may not be able to fly.

Now, we add new information to the DB, "Tweety can fly (Fly(tweety))". Then $\llcorner$Fly$\lrcorner$ is $\lambda x.(x = \text{tweety})$ and $\ulcorner$Fly$\urcorner$ is unchanged. Under this circumstance, we cannot obtain either '$\lambda x.(\text{Bird}(x) \wedge \llcorner\text{Fly}\lrcorner(x)) >> \lambda x.(\text{Bird}(x) \wedge \neg\ulcorner\text{Fly}\urcorner(x))$' or '$\lambda x.(\text{Bird}(x) \wedge \neg\ulcorner\text{Fly}\urcorner(x)) >> \lambda x.(\text{Bird}(x) \wedge \llcorner\text{Fly}\lrcorner(x))$'. Hence, (E1.4) is not a theorem of the DB any more. However, if the DB also knows "Jack can fly (Fly(jack))", the theorems will change more dramatically. In this case, using (E1.2) $\lambda x.(\text{Bird}(x) \wedge \llcorner\text{Fly}\lrcorner(x)) >> \lambda x.(\text{Bird}(x) \wedge \neg\ulcorner\text{Fly}\urcorner(x))$ follows. Therefore, the directivity of 'Bird' changes and using Directional-G(A $\wedge$ Fly(tweety) $\wedge$ Fly(jack); Fly / Bird),

$$\forall x.(\text{Bird}(x) \supset (\text{Fly}(x) \equiv \neg x = \text{p-suke}) \qquad \text{(E1.4)}$$

is obtained. This means "P-suke is the only bird that cannot fly," P-suke comes to be considered to be abnormal with respect to flying inside the world of birds.

4. An Application to *is-a* Hierarchy and Consideration on a First Order Formulation

Consider a simple example of an is-a hierarchical system.

Example 2.

Let A be

$(\alpha \Rightarrow \beta \supset \forall x.(\alpha(x) \supset \beta(x)) \wedge \beta >> \alpha)$, for all predicate α and β, (E2.1)

$\wedge$ Sparrow $\Rightarrow$ Bird (E2.2)

$\wedge$ Penguin $\Rightarrow$ Bird (E2.3)

$\wedge$ Bird $\Rightarrow$ Animate (E2.4)

$\wedge$ Reptilian $\Rightarrow$ Animate (E2.5)

$\wedge \neg\exists x.(\text{Sparrow}(x) \wedge \text{Penguin}(x))$ (E2.6)

$\wedge\ \neg\exists x.(\mathrm{Bird}(x) \wedge \mathrm{Reptilian}(x))$ (E2.7)

$\wedge\ \forall x.(\mathrm{Sparrow}(x) \supset \mathrm{Fly}(x))$ (E2.8)

$\wedge\ \forall x.(\mathrm{Penguin}(x) \supset \neg\mathrm{Fly}(x))$ (E2.9)

$\wedge\ \forall x.(\mathrm{Reptilian}(x) \supset \neg\mathrm{Fly}(x))$ (E2.10)

$\wedge\ \mathrm{Sparrow} > > \mathrm{Penguin}$ (E2.11)

$\wedge\ \mathrm{Reptilian} > > \mathrm{Bird},$ (E2.12)

where the intended meaning of binary relation $\Rightarrow$ is '*is-a*'.

Now, let us consider the general rules with respect to Fly inside each class of this hierarchical system. As we define the interpretation of Fly using directional generalization rule inside each class, the obtained definition of Fly inside a class must influence definition of Fly inside other classes which are obtained later. We define Fly from lower class to upper (from leaves to root), then we can obtain natural results. In this example, if we define Fly inside Animate previously to inside Bird, we obtain

$$\forall x.\ (\mathrm{Animate}(x) \wedge \neg\mathrm{Sparrow}(x) \supset \mathrm{Fly}(x)). \qquad \text{(E2.14)}$$

However, rather than (E2.15), we prefer

$$\forall x.\ (\mathrm{Animate}(x) \wedge \neg(\mathrm{Bird}(x) \wedge \neg\mathrm{Penguin}(x)) \supset \mathrm{Fly}(x)) \qquad \text{(E2.15)}$$

as a result in which we generalize a rule with respect to Fly inside Bird previously and then using the rule obtained we generalize it inside Animate. Hence, we must use the directional generalization rule in the following way:

$$\text{Directional-G(Directional-G(} \cdots \text{Directional-G(A; P / } C_1) \cdots ; \text{P / } C_{n-1}); \text{P / } C_n), \qquad \text{(O1.1)}$$

which is denoted by Directional-G(A; P / C_1, $\cdots$, C_{n-1}, C_n). That is,

$$\text{Directional-G}(A; P / C_1, \cdots, C_{n-1}, C_n) = \text{Directional-G}(Gn; P / Cn), \qquad (O1.2)$$

where $G_0 = A$, $G_i = \text{Directional-G}(G_{i-1}; P / C_{i-1})$ and $\neg(C_i \Rightarrow C_j)$ $(1 \leqq j \leqq i)$. We refer to Directional-G(A; P / $C_1, \cdots, C_{n-1}, C_n$) as the *prioritized directional generalization rule.*

An unsatisfiable point of this formulation is on computational aspects. Directional-G(Gi; P / Ci) is expressed by second order logic and unfortunately Directional-G(A; P / C1, ⋯ , Cn-1, Cn) become more complicated. Then, let us try to collapse these formula ions into first order logic, giving some constraints to given formulas. The basic idea is this: the essential reason for second order logic formulation is that there are some candidate properties of $\llcorner P \lrcorner$ (or $\ulcorner P \urcorner$) based on their minimal (or maximal) models with respect to P. Therefore, if there exists the minimum (or maximum) model (considering logically equivalent models as one model), there exists the only candidate of $\llcorner P \lrcorner$ (or $\ulcorner P \urcorner$) and we can handle $\llcorner P \lrcorner$ (or, $\ulcorner P \urcorner$) as a constant property. That is, we assume that a set of entities which we know satisfy P corresponds to a minimum extension of P which satisfy given facts. Now we revise our formulation, (M1), (M2) with respect to $\llcorner P \lrcorner$ and $\ulcorner P \urcorner$ in this sense as follows:

$$A[\llcorner P \lrcorner] \wedge \forall p.\,(A[p] \supset \forall x.\,(\llcorner P \lrcorner(x) \supset p(x))) \qquad (M1')$$

and

$$A[\ulcorner P \urcorner] \wedge \forall p.\,(A[p] \supset \forall x.\,(p(x) \supset \ulcorner P \urcorner(x))). \qquad (M2')$$

We refer to such a predicate, $\llcorner P \lrcorner$, as a *minimum candidate predicate* (abbreviated *min.c.p.*) *of* P *in* A and to $\ulcorner P \urcorner$ as a *maximum candidate predicate* (abbreviated *max.c.p.*) *of* P *in* A.

Now, let us consider the maximum model in more detail.

Proposition 2 (logically uniqueness of min.c.p. and max.c.p.).

On the premise of a given formula, A, for any predicates, ρ1 and ρ2,

1) $A[\rho 1] \wedge \forall p.\,(A[p] \supset \forall x.\,(\rho 1(x) \supset p(x)))$

$\wedge A[\rho 2] \wedge \forall p.\,(A[p] \supset \forall x.\,(\rho 2(x) \supset p(x)))$

$\quad \supset \forall x.\,(\rho 1(x) \equiv \rho 2(x))$ (P2.1)

2) $A[\rho 1] \wedge \forall p.\,(A[p] \supset \forall x.\,(p(x) \supset \rho 1(x)))$

$\wedge A[\rho 2] \wedge \forall p.\,(A[p] \supset \forall x.\,(p(x) \supset \rho 2(x)))$

$\quad \supset \forall x.\,(\rho 1(x) \Leftarrow \rho 2(x))$ (P2.2)

Proof. By predicate calculus.

Proposition 2 says that if there exist a predicate that satisfy (M1') (or (M2')) any predicates satisfying (M1') (or (M2')) is logically equivalent to the predicate.

Theorem 1.a.

On the premise of a given formula, A, for any property, $\llcorner P \lrcorner$, in which no predicate in P occurs,

1) $\forall x.\,(\llcorner P \lrcorner(x) \supset P(x)) \supset \forall p.\,(A[p] \supset \forall x.\,(\llcorner P \lrcorner(x) \supset p(x)))$

holds, and

2) there exists a minimum model of A with respect to P if there exists $\llcorner P \lrcorner$ such that $A[\llcorner P \lrcorner] \wedge \forall x.\,(\llcorner P \lrcorner(x) \supset P(x))$ holds.

3) $\llcorner P \lrcorner$ is a min.c.p. if $A[\llcorner P \lrcorner] \wedge \forall x.\,(\llcorner P \lrcorner(x) \supset P(x))$ holds.

Proof.

1) Assume $A \vdash \forall x.\,(\llcorner P \lrcorner(x) \supset P(x)) \wedge A[p]$ for some $\llcorner P \lrcorner$ and p. Using Kleene's theorem [1], $A[p] \vdash \forall x.\,(\llcorner P \lrcorner(x) \supset P(x))[p]$. Here, no predicate in P occurs in $\llcorner P \lrcorner$, therefore, $\llcorner P \lrcorner[p] = \llcorner P \lrcorner$. So, $A[p] \vdash \forall x.\,(\llcorner P \lrcorner(x) \supset p(x))$. From this and $A \vdash A[p]$, $A \vdash \forall x.\,(\llcorner P \lrcorner(x) \supset p(x))$.

2) Using this assumption and 1), $A[\llcorner P \lrcorner] \wedge \forall p.\,(A[p] \supset \forall x.\,(\llcorner P \lrcorner(x) \supset p(x))$ follows. A model M in which the extension of P is equivalent to the extension of $\llcorner P \lrcorner$ is minimal with respect to P for all models.

3) Using this assumption and 1), and predicate calculus.

Theorem 1.b.

On the premise of a given formula A, for any property, $\ulcorner P \urcorner$, in which no predicate in P occurs,

1) $\forall x.\,(P(x) \supset \ulcorner P \urcorner(x)) \supset \forall p.\,(A[p] \supset \forall x.\,(p(x) \supset \ulcorner P \urcorner(x)))$

holds and

2) there exists a maximum model of A with respect to P if there exists $\ulcorner P \urcorner$ such that $A[\ulcorner P \urcorner] \wedge \forall x.\,(P(x) \supset \ulcorner P \urcorner(x))$ holds..

3) $\ulcorner P \urcorner$ is a max.c.p. if $A[\ulcorner P \urcorner] \wedge \forall x.\,(P(x) \supset \ulcorner P \urcorner(x))$ holds.

Proof. Similarly for **Theorem 1.a.**

If we restrict $\llcorner P \lrcorner$ and $\ulcorner P \urcorner$ to predicates such that no predicate in P occurs in the predicates, we can leave out second order formulas $\forall p.\,(A[p] \supset \forall x.\,(\llcorner P \lrcorner(x) \supset p(x))$ and $\forall p.\,(A[p] \supset \forall x.\,(p(x) \supset \ulcorner P \urcorner(x))$ from (M1') and (M2') using **Theorem 1**, and we can also omit the second order quantifiers in (D2) using **Proposition 2**. Therefore, our new formulation, *first order directional generalization rule*, is

$$
\begin{aligned}
&A[P] \\
&\wedge (\lambda x.(\,K(x) \wedge \llcorner P \lrcorner(x)) >> \lambda x.(\,K(x) \wedge \neg \ulcorner P \urcorner(x)) \supset \forall x.(K(x) \supset (P(x) \equiv \ulcorner P \urcorner(x)))) \\
&\wedge (\,\lambda x.(\,K(x) \wedge \neg \ulcorner P \urcorner(x) >> \lambda x.(\,K(x) \wedge \llcorner P \lrcorner(x)) \supset \forall x.(K(x) \supset (P(x) \equiv \llcorner P \lrcorner(x)))) \\
&\quad),
\end{aligned}
\tag{D3}
$$

where $\llcorner P \lrcorner$ is min.c.p. and $\ulcorner P \urcorner$ is max.c.p. in which no predicate in P occurs.

It can be easily shown to there exist such $\llcorner P \lrcorner$ and $\ulcorner P \urcorner$ for a certain class of given formulas. Next, we show this.

Definition [*independently separable formula*].

A formula, A, is an *independently separable formula of* P if A can be transformed into the following form,

$$U \wedge \forall x.(L(x) \supset P(x)) \wedge \forall x.(P(x) \supset G(x)), \tag{D4}$$

where no predicate in P occurs in U, L(x) or G(x).

Proposition 4.

If a formula, A, is a independently separable formula of P, there exist a min.c.p. and a max.c.p.in which no predicate in P occurs.

Proof. Let A be equivalent to $U \wedge \forall x.(L(x) \supset P(x)) \wedge \forall x.(P(x) \supset G(x))$. Then L is a min.c.p. and G is a max.c.p. in which no predicate in P occurs..

Proposition 5.

If both A and B are independently separable formulas, $A \wedge B$ is also an independently separable formula.

Proof. By predicate calculus.

Now, if a given formula, A, is an independently separable formula, by the way shown in the proof of **Proposition 4**, both a min.c.p. and a max.c.p in which no predicate in P occur are easily obtained. And, moreover, the first order directional generalization rule is also an independently separable formula by **Proposition 5**. Therefore, the prioritized directional generalization rule can be collapsed into first order logic.

Example 2 (continued).

From the definition of prioritized directional generalization rule,

$$\text{Directional-G(A; Fly / Bird, Animate)} \tag{E2.16}$$

$$\equiv \text{Directional(Directional-G(A; Fly / Bird); Fly / Animate)} \tag{E2.17}$$

Here, A is an independently separable formula. So, we can easily find that a min.c.p. of Fly in A is Sparrow and a max.c.p. of Fly in A is $\lambda x.\neg(\text{Penguin}(x) \vee \text{Reptilian}(x))$ by the way shown in the proof of **Proposition 4**. Therefore,

$$\text{Directional-G(A; Fly / Bird)} \tag{E2.18}$$

$$\equiv$$

$$
\begin{aligned}
&A \\
&\wedge (\lambda x.(Bird(x) \wedge Sparrow(x)) >> \lambda x.(Bird(x) \wedge (Penguin(x) \vee Reptilian(x))) \\
&\quad \supset \forall x.(Bird(x) \supset (Fly(x) \equiv \neg(Penguin(x) \vee Reptilian(x)))) \\
&\wedge (\lambda x.(Bird(x) \wedge (Penguin(x) \vee Reptilian(x))) >> \lambda x.(Bird(x) \wedge Sparrow(x)) \\
&\quad \supset \forall x.(Bird(x) \supset (Fly(x) \equiv Sparrow(x)))) \qquad \text{(E2.19)} \\
\equiv \quad &A \wedge \forall x.(Bird(x) \wedge \neg Penguin(x) \supset Fly(x)) \qquad \text{(E2.20)}
\end{aligned}
$$

Similarly, a min.c.p. of Fly in Directional-G(A; Fly / Bird) is $\lambda x.(Bird(x) \wedge \neg Penguin(x))$ and a max.c.p. of Fly is $\lambda x.\neg(Penguin(x) \vee Reptilian(x))$. Therefore,

$$
\begin{aligned}
&\text{Directional-G(A; Fly / Bird, Animate)} \qquad \text{(E2.21)} \\
&\equiv \\
&\quad A \\
&\wedge \forall x.(Bird(x) \wedge \neg Penguin(x) \supset Fly(x)) \\
&\wedge \forall x.(Animate(x) \wedge \neg(Bird(x) \wedge \neg Penguin(x)) \supset \neg Fly(x)) \qquad \text{(E2.22)}
\end{aligned}
$$

4. Conclusion and Remarks

In the example 2, assume that all that we know about foo is that foo is a bird and we want to know whether foo can fly or not. In this case, we may need introduce predicates allowed to vary in our formulation like parallel circumscription[2]. Other solution without introducing such predicates is to add

$\lambda x.(x = foo) \Rightarrow Bird$

and necessary formulas which expresses that $\lambda x.(x = foo)$ is disjoint to some other classes like (E2.6).

Partially directional circumscription can be easily extended to its more general version based on formula circumscription.

A[P]

$$\wedge \forall \rho.(A[\rho] \wedge \forall x.((\Phi(x) \supset ((E(x) \equiv \Lambda(x)) \supset (E[\rho](x) \equiv \Lambda(x))))$$
$$\supset \forall x.(\Phi(x) \supset (E(x) \equiv E[\rho](x))))),$$

where wff E means some concepts, expressing the property which we want to generalize.

We hope this research extends the sphere of interest of researchers in non-monotonic reasoning and serves as a new stimulus to machine learning, analogy and inductive inference.

ACKNOWLEDGMENTS

I would like to thank Dr. Koichi Furukawa and Mr. Hirohisa Seki for their useful comments. Also, I wish to express my gratitude to Dr. Kazuhiro Fuchi, Director of the ICOT Research Center, who provided me with the opportunity to pursue this research.

REFERENCES

[1] Kleene,S.C.: *Introduction to Metamathematics*, North-Holland, 1971, CH. Ⅶ.

[2] Lifschitz,V.: Computing circumscription, in: *Proceedings of Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA (1985) 121-127.

[3] McCarthy,J.: Circumscription - a form of non-monotonic reasoning, *Artificial Intelligence* 13 (1980) 27-39.

[4] McCarthy,J.: Application of circumscription to formalizing common-sense knowledge, *Artificial Intelligence* 28 (1986) 89-116.

[5] Cohen,P.R. and Feigenbaum E.A.: *The Handbook of Artificial Intelligence*, Vol. 3.