A Simple Method of Mutual Translation between Japanese Sentences and Horn Clauses

Inoue, Hitoshi
Computation Center, Kyushu University

Arikawa, Setsuo
Research Institute of Fundamental Information Science Kyushu University

Takeya, Shun-ichi
Computation Center, Kyushu University

出版情報：RIFIS Technical Report. 45, 1991-09-06. Research Institute of Fundamental Information Science, Kyushu University

RIFIS-TR-CS-45

RIFIS Technical Report

A Simple Method of Mutual Translation between Japanese Sentences and Horn Clauses

Hitoshi Inoue
Setsuo Arikawa
Shun-ichi Takeya

September 6, 1991

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

E-mail: jin@keisu.kyushu-u.ac.jp Phone: 092 (641)1101 Ex. 5982

A simple method of mutual translation between Japanese sentences and Horn clauses

Hitoshi Inoue †
jin@keisu.kyushu-u.ac.jp

Setsuo Arikawa ‡
arikawa@rifis.sci.kyushu-u.ac.jp

Shun-ichi Takeya †
shun@keisu.kyushu-u.ac.jp

† Computation Center, Kyushu University 36, Fukuoka 812, Japan
‡ Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan

Abstract

This paper presents a simple method of mutual translation between Japanese sentences and Horn clauses, which has been implemented in our analogical reasoning system as a man-machine interface. Natural language interface systems generally need complicated semantic processing using a large dictionary. However, such processing is not suitable for our interface system because the interface overhead becomes too large in comparison with the main reasoning processing and the vocabulary is limited by the dictionary. From this viewpoint, we adopt a textual processing. Our system does not have a large dictionary, but instead we put some restrictions on sentences inputted to the system; for example, each sentence should be separated into words by spaces and be expressed uniformly. We also present methods which keep input Japanese sentences natural and which remove ambiguities by the interaction between users and the system.

1. Introduction

In order to make computers assist us in intellectual information processing, human knowledge must be represented in computer-understandable form. A set of Horn clauses (Prolog program), one of typical knowledge representation languages, runs efficiently on computers, and is easy to represent human knowledge. However, Horn clauses are not always easy to understand for those people who are unfamiliar with clausal notations.

On the other hand, natural languages are easy for people to understand, but are hard for computers to process.

In this paper, to assist users of our analogical reasoning system ARTS [1, 2] in input and output processing, we propose a simple Japanese language which has some restrictions but is easy to understand. We also consider a method of mutual translation between sentences of the language and Horn clauses. Of course this method is applicable to interface systems for other knowledge information processing systems which use Horn clauses as inner representations.

Natural language interface systems generally have a large knowledge base about the grammar and words of the language, and translate input sentences into inner representations using the knowledge [3]. However we consider that proper interface systems should be based on the following policy.

(1) We should distinguish the following two things. One is that an interface system has general knowledge about natural language. Another is that the system accepts sentences of the language.

(2) Complicated semantic processing causes the overhead large, and does not work efficiently as an interface system.

(3) A large dictionary is not useful if inputted words are not stored in it. Users should be able to define any term and any predicate.

Horn clauses are classfied into facts, rules and questions. Therefore the corresponding Japanese sentences are naturally restricted. From this viewpoint, we adopt a textual processing, namely analyzing input sentences without a large dictionary. Our system does not have a dictionary, but instead we put some restrictions on input sentences such as each sentence must be separated into words by spaces and be expressed uniformly. However we have a very small dictionary of basic words such as postpositional paritcles and conjunctions, because these words are indispensable for knowledge representation and make translation efficient. By considering simple rules of Japanese grammar which can be treated textually, we keep input Japanese sentences natural.

Thus our interface system is not to understand natural languages, but to assist users in input and output processing.

Section 2 gives a method of translation from simple Japanese sentences into Horn clauses. Section 3 discusses a method which keeps input Japanese sentences natural from the logical viewpoint. Section 4 discusses a method which removes ambiguities from input sentences by the interaction between users and the system. Section 5 gives a method of translation from Horn clauses into Japanese sentences.

2. Translation from Japanese sentences into Horn clauses

Let A, B_i be atomic formulas(atoms). Then, Horn clauses are classfied as follows:

(1)	$A \leftarrow$	(fact)
(2)	$A \leftarrow B_1, \cdots, B_n$	(rule)
(3)	$\leftarrow B_1, \cdots, B_n$	(question)

The corresponding Japanese sentences are basically as follows respectively.

		(meaning in English)
(1)	Aである	(A)
(2)	もしB_1であって$\cdots B_n$であるならば, Aである	(*If* $B_1, \cdots$ *and* B_n *then* A)
(3)	B_1であって$\cdots B_n$であるか	($B_1, \cdots$ *and* B_n *?*)

2.1 Translation of facts

First we consider a translation from a Japanese sentence for a fact into an atom, which is a basic element of a clause. The corresponding Japanese sentence for a fact is a simple sentence. The predicate of the sentence or the word representing relationship should be seletected as the predicate symbol, and words representing things should be selected as terms. But we can not decide whether each word in the sentence represents a relationship or a thing, because we have no dictionaries about predicates and things. Japanese sentences feature the following, so we treat only those sentences that satisfy the features:

(1) A typical Japanese word order is 'subject+objects+verb'.

(2) A subject or objects have a particle at the postposition of them.

Following these feautres, we can translate a Japanese sentence into a Horn clause without a large dictionary, by regarding the word at verb position and words at subject or object position as a predicate symbol and terms respectively.

Then the translation is as follows:

	項$_1$ 助詞$_1 \cdots$ 項$_n$ 助詞$_n$ 述語	($Term_1$ $Particle_1 \cdots$ $Term_n$ $Particle_n$ $Predicate$)
$\Longrightarrow$	述語 (項$_1, \cdots,$ 項$_n$)	($Predicate(Term_1, \cdots, Term_n)$)

2.2 Translation of rules

A Japanese sentence for a rule is in the following form:

		(meaning in English)
(1)	もし $B_1 \cdots B_n$ ならば A	(*If* $B_1, \cdots$ *and* B_n *then* A)
(2)	$B_1 \cdots B_n$ ので A	(*Since* $B_1, \cdots$ *and* B_n, A)
(3)	A なぜなら $B_1 \cdots B_n$ ので	(A *because* $B_1, \cdots$ *and* B_n)
		(A, B_i : simple sentences)

Let A' and B_i' be translations from simple sentences A and B_i respectively. Then the translation is as follows:

$$A' \leftarrow B_1', \cdots, B_n'$$

2.3 Translation of questions

A Japanese sentence for a question is in the following form:

$B_1 \cdots B_n$ か $(B_1, \cdots \textit{ and } B_n\ ?)$

$(B_i :$ a simple sentence$)$

This is a sequence of simple sentences which ends with the particle 'か'('か' means '?').

Let B_i' be a translation from a simple sentence B_i. Then the translation is as follows:

$$\leftarrow B_1', \cdots, B_n'$$

2.4 A simple Japanese grammar

Fig.1 shows the simple Japanese grammar for translation into Horn clauses.

The interface system parses input sentences not character by character but string by string separated by spaces or punctuation marks because of the following reasons:

(1) Character by character parsing is ambiguous because the system has no dictionaries except the words in **Fig.1**.

(2) Word by word Parsing is more efficient than that of character by character.

For natural expression, spaces may be omitted between ⟨TERM⟩ and ⟨PARTICLE⟩, and between ⟨PRED⟩ and ⟨THEN⟩.

<table>
<tr><td>⟨FACT⟩</td><td>::=</td><td>⟨ATOM⟩</td></tr>
<tr><td>⟨RULE⟩</td><td>::=</td><td>⟨IF⟩ ⟨ATOMS⟩ ⟨THEN⟩ ⟨ATOM⟩ | ⟨ATOMS⟩ ⟨SO⟩ ⟨ATOM⟩ |
⟨ATOM⟩ ⟨BECAUSE⟩ ⟨ATOMS⟩ ⟨SO⟩</td></tr>
<tr><td>⟨QUERY⟩</td><td>::=</td><td>⟨ATOMS⟩ か</td></tr>
<tr><td>⟨ATOMS⟩</td><td>::=</td><td>⟨ATOM⟩ | ⟨ATOM⟩ ⟨AND⟩ ⟨ATOMS⟩</td></tr>
<tr><td>⟨ATOMS⟩</td><td>::=</td><td>⟨PRED⟩ | ⟨TERMS⟩ ⟨PRED⟩</td></tr>
<tr><td>⟨TERMS⟩</td><td>::=</td><td>⟨TERM⟩ | ⟨TERM⟩ ⟨TERMS⟩</td></tr>
<tr><td>⟨TERM⟩</td><td>::=</td><td>⟨THING⟩ ⟨PARTICLE⟩</td></tr>
<tr><td>⟨THING⟩</td><td>::=</td><td>any string</td></tr>
<tr><td>⟨PRED⟩</td><td>::=</td><td>any string</td></tr>
<tr><td>⟨PARTICLE⟩</td><td>::=</td><td>は | が | の | に | を | へ | と | から | より | で</td></tr>
<tr><td>⟨AND⟩</td><td>::=</td><td>ε | そして | かつ | しかも</td></tr>
<tr><td>⟨IF⟩</td><td>::=</td><td>ε | もし | もしも</td></tr>
<tr><td>⟨THEN⟩</td><td>::=</td><td>ならば | なら</td></tr>
<tr><td>⟨SO⟩</td><td>::=</td><td>ので | から</td></tr>
<tr><td>⟨BECAUSE⟩</td><td>::=</td><td>なぜなら | というのは</td></tr>
<tr><td></td><td></td><td>(ε : empty word)</td></tr>
</table>

Fig.1 A simple Japanese grammar for translation into Horn clasues

Example 1 The sentence'Xは　Yの　子供である. なぜなら　Yは　Xの　父親であるから' (X *is-child-of* Y *because* Y *is-father-of* X) satisfies the above grammar, and is translated into the following:

子供である $(X, Y) \leftarrow$ 父親である (Y, X)　(*is-child-of* $(X, Y) \leftarrow$ *is-father-of* (Y, X))

Note here that the predicate symbols are the literal copies of the words in input sentences.

We think the restriction that sentences should be separated by spaces is not so strong because we generally use those expressions to make the meaning of sentences clear. We also think the restriction that sentences should be expressed uniformly is not so strong because it is necessary for clausal notations.

Japanese sentences for rules are expressed in the above three forms basically, but are expressed in various ways such as omitting 'もし'(*if*), or using 'というのは'(*for*) instead of 'なぜなら'(*because*). We make sentences for rules more expressive by defining synonyms such as considering ⟨IF⟩ as nonterminal symbol in **Fig.1**.

In the implementation, the grammar and the translation are expressed by a DCG [4].

3. Keeping input sentences natural

In the former section, we considered a simple Japanese grammar and a translation into Horn clauses. However, a sentence which satisfies the grammar is not always natural and not translated as we expect. So we consider a method which keeps input sentences natural and translates them as we expect.

3.1 Conjugation of predicates

The expression '〜である そして 〜である ···' (*predicate1, predicate2, and* ···), a conditional part of Japanese sentences for rules, is not natural. We express it as '〜であって (そして) 〜であって ···', (*predicate1', predicate2',* ···), the conjugation form of '〜である', which is used to link sentences and means '*and*'. We call this conjugation form *te*-form because the conjugation form ends with 'て' ('*te*'). In this notation, the *te*-form predicate is regarded as a different predicate against our expectation. So we consider a tranformation from the *te*-form into the original form.

te-form	original form
p って	p う, p つ, p る, p く
p んで	p ぬ, p ぶ, p む
p いて	p く, p いる
p して	p する, p す
p くて	p い
p きて	p くる
p て	p る
p いで	p ぐ
p で	p だ

p : the prefix of a predicate

Table 1 The conjugation of predicates

Table 1 shows the transformation from the *te*-form of Japanese predicate into the original form, that is, a word in first column is transformed into word(s) in second column textually. Here the problem is the treatment if the transformation does not decide the original form uniformly. For example, assume that a input sentence contains the predicate '父親であって' ('*chichioya-deatte*'). Since the suffix of the predicate is 'って' ('*tte*'), it can be transformed into either '父親であう' ('〜*deau*'), '父親であつ' ('〜*deatsu*'), '父親である' ('〜*dearu*'), or '父親であく' ('〜*deaku*') by **Table 1**. But if a sentence containig the predicate '父親である' ('〜*dearu*') is inputted later, the orignal of '父親であって' ('〜*deatte*') is decided to '父親である' ('〜*dearu*').

This transformation is effective when the original form of a *te*-form predicate appears somewhere in inputted sentences. Thus we only consider the case when the original form of a predicate is not inputted.

The purpose to decide the original form of a predicate is to identify the *te*-form with the original form. Assume the original form of a *te*-form predicate in a goal clause is undecided. The undecided *te*-form predicate only appears in bodies of clauses becasue predicates in heads of clauses are always in original form. This means the undecided predicate is not defined in the clauses, and the goal containing it must fail. After all the original form of a predicate does not need to be decided because the goal fails regardless of it.

Thus in case that inputted Japanese sentences finally run as a Prolog program, the original form of a predicate is always decided when the predicate is used later.

Example 2 Assume the following Japanese sentences be inputted.

	(meaning in English)
もし Xは Yの 父親であって	(*If X is-father-of Y and*
Yは Zの 親であるならば,	*Y is-parent-of Z then*
Xは Zの 祖父である.	*X is-grandfather-of Z.*)
メアリは ジョンの 親である.	(*Mary is-parent-of John.*)
トムは ジョンの 祖父であるか.	(*Is Tom grandfather-of John?*)

These are translated into as follows:

祖父である (X, Z)	(*is-grandfather-of*(X, Z)
$\leftarrow$ 父親であって (X, Y), 親である (Y, Z)	$\leftarrow$ *is-father-of-and* (X, Y),*is-parent-of*(Y, Z))
親である (メアリ, ジョン) $\leftarrow$	(*is-parent-of*(*mary,john*) $\leftarrow$)
$\leftarrow$ 祖父である (トム, ジョン)	($\leftarrow$ *is-grandfather-of*(*tom,john*))

Then the goal '$\leftarrow$ 祖父である (トム, ジョン)' ('$\leftarrow$ *is-grandfather-of*(*tom,john*)') is tried. By the definition of the predicate '祖父である' ('*is-grandfather*'), the goal '$\leftarrow$ 父親であって (トム,Y), 親である (Y, ジョン)' ('$\leftarrow$ *is-father-of-and*(*tom*, Y), *is-parent-of*(Y,*john*)') is selected next. Since the predicate '父親である' ('*is-father-of*') is not defined in the above clauses, the subgoal '$\leftarrow$ 父親であって (トム,Y)' ('$\leftarrow$ *is-father-of-and*(*tom*, Y)') fails. Even if the *te*-form '父親であって' ('*is-father-of-and*') is expressed as the original form '父親である' ('*is-father-of*'), the subgoal fails. So the original form of the predicate '父親であっ

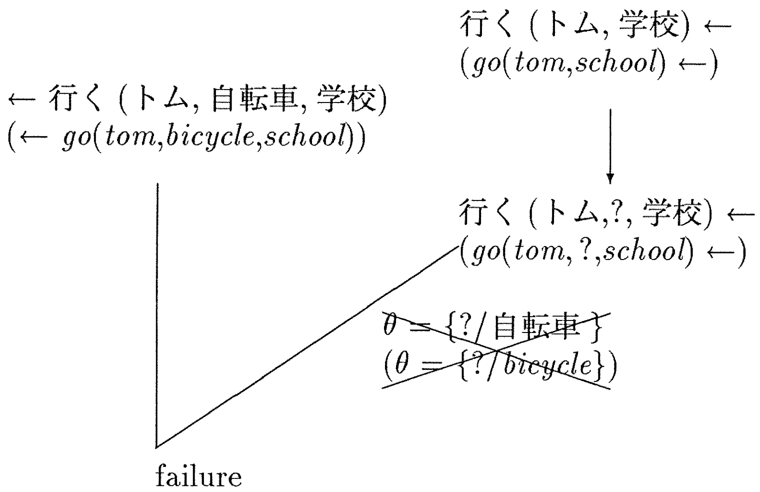
て' ('*is-father-of-<u>and</u>*') does not need to be decided.

3.2 Arguments in predicate symbols

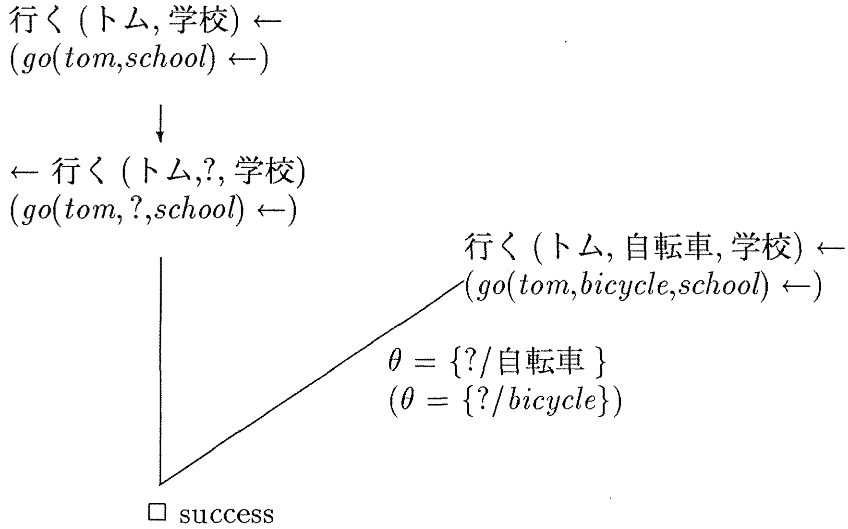
Most Prolog systems distinguish the same predicate symbol with different arities, and unify two atoms or terms in the same position. However, in Japanese sentences the word order of objects(terms) is relatively free, and some of them is omitted.

So we consider a method which exchanges the positions of arguments and to add missing arguments dynamically. Since the Japanese sentences considered here have a particle following each term, we adopt the following method: For every input predicate the system stores the information of the particles of the arguments. When a sentence containing the predicate is inputted, the system exchanges the positions of arguments using the stored information.

We consider the two cases. The first case is that the atom in the head of a clause changes. By '?' we denote an added argument. Assume that the sentence 'トムは 学校に 行く' ('*Tom goes to school*') is already stored as '行く(トム,学校) ←' ('$go(tom,school) \leftarrow$'). Now assume that the goal sentence 'トムは 自転車で 学校に 行くか' ('*Does Tom go to school by bicycle?*') is inputted. Then the former fact is changed into '行く(トム,?,学校) ←' ('$go(tom,?,school) \leftarrow$'). Tom goes to school by something, but we do not know whether he goes there by bicycle. Therefore the goal should fail. If we regard '?' as a variable, the goal is unifiable by the substitution {?/自転車} ($\{?/bicycle\}$) and succeed. Therefore we should regard '?' as a constant.



The second case is that the atom in the body of a clause changes. Assume that the sentence 'トムは 自転車で 学校に 行く' ('*Tom goes to school by bicycle*') is already stored and the goal sentence 'トムは 学校に 行くか' ('*Does Tom go to school?*') is inputted next. The latter sentence just means whether Tom goes to school or not and the means by which he goes there is not important. So the goal should succeed by the substitution {?/自転車 } ({?/*bicycle*}). Therefore we should regard '?' as a variable.



The above observation is reasonable from the following logical viewpoint. Assume a new argument of an atom A be added. It is considered that there exists something, so we replace the atom A for $\exists X A(X)$ where X is a new variable that does not appear in the clause, and transform this into a Skolem standard form [5].

(a) The case the atom A is in the head

We consider the clause $A \leftarrow B$.

$$
\begin{array}{lll}
 & A \leftarrow B & \\
\Longrightarrow & \exists X A(X) \leftarrow B & \text{(replacement)} \\
\Longleftrightarrow & \forall(\exists X A(X) \vee \sim B) & \text{(logically equivalent)} \\
\Longleftrightarrow & \forall(\exists X (A(X) \vee \sim B)) & \text{(logically equivalent)} \\
\Longrightarrow & \forall(A(a) \vee \sim B) & \text{(Skolem standard form)}
\end{array}
$$

Therefore the clause is transformed into $A(a) \leftarrow B$ where 'a' is a Skolem constant.

(b) The case the atom A is in the body

We consider the clause $B \leftarrow A$.

$$
\begin{array}{lll}
 & B \leftarrow A & \\
\Longrightarrow & B \leftarrow \exists X A(X) & \text{(replacement)} \\
\Longleftrightarrow & \forall(B \vee \sim \exists X A(X)) & \text{(logically equivalent)} \\
\Longleftrightarrow & \forall(B \vee \forall X \sim (A(X))) & \text{(logically equivalent)} \\
\Longleftrightarrow & \forall(B \vee \sim A(X)) & \text{(logically equivalent, } X \text{ does not appear in } B\text{)}
\end{array}
$$

Therefore the clause is transformed into $B \leftarrow A(X)$ where X is a new variable that does not appear in the clause $B \leftarrow A$.

4. Removing ambiguities

The Japanese grammar for translation (**Fig.1**) is ambiguous because the word 'から' can be interpreted as both ⟨PARTICLE⟩ and ⟨SO⟩. We can change the grammar to remove such an ambiguity, but it is not suitable because translatable Japanese sentences will be restricted.

The translation considered here is for an interface system and Japanese sentences are inputted through the interaction with the user. Thus we adopt a method that the system asks the user again in case the input sentence is ambiguous.

Example 3

|:<u>トムは 昨日から 車を 運転している</u>	|:<u>*Since yesterday Tom has-been-driving his-car*</u>
入力された文は曖昧です	*The input sentence is ambiguous*
1 トムは 昨日から 車を 運転している	1 *Tom has-been-driving his-car since yesterday*
[運転している (トム, 昨日, 車) ←]	[*has-been-driving*(*tom*,*his-car*,*yesterday*) ←]
2 もし, トムは 昨日ならば, 車を 運転している	2 *If yesterday Tom, has-been-driving his-car*
[運転している (車) ← 昨日 (トム)]	[*has-been-driving*(*his-car*) ← *yesterday*(*tom*)]
どちらを表していますか? <u>1</u>	*Which do you mean?* <u>1</u>

(Underlined sentences are the user's input.)

5. Translation from Horn clauses into Japanese sentences

In this section, we consider a method which translates Horn clauses into Japanese sentences reversely. We Japanese can translate the fact '行く (トム, バス, 学校) ←' ('*go*(*tom*,*bus*,*school*) ←') into the Japanese sentence 'トムはバスで学校に行く'('*Tom goes to school by bus*') quite easily because we know the following:

(1) the cases of Japanese predicates,

(2) the cases of the arguments.

Since our system does not have such knowledge about cases of predicates, it seems impossible to translate Horn clauses into Japanese sentences. However, the Horn clauses to be translated *into* Japanese sentences were already translated *from* Japanese sentences at the stage of storing the knowledge. Therefore the system knows the above (1) and (2), and can easily translate them into Japanese sentences.

6. Conclusion

We considered a simple method of mutual translation between Japanese sentences and Horn clauses to assist users of our analogical reasoning system which uses Horn clauses as inner representations in input and output processing. This method features that the system does not have a large dictionary and it processes input Japanese sentences textually. Although the method restricts translatable Japanese sentences, the method is natural and suitable for exact knowledge representation and fast processing.

We implemented the Japanese interface system in our analogical reasoning system ARTS on SUN-3 workstation using K-Prolog based on the method we have discussed in this paper.

References

[1] Haraguchi,M. : Towards a Mathematial Theory of Analogy, Bulletin of Information and Cybernetics, Vol.21(1985), pp.29-56

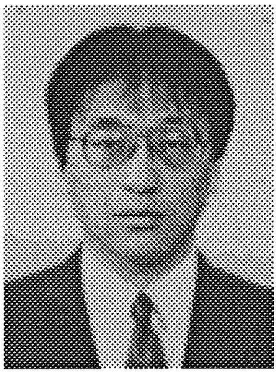
[2] Haraguchi,M. : Analogical Reasoning Using Transformation of rules, Bulletin of Information and Cybernetics, Vol.22(1986), pp.1-8

[3] Tennant,H. : Natural Language Processing, Petrocelli Books, Inc.(1981)

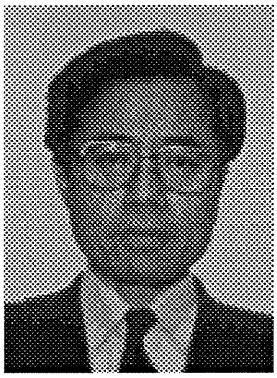
[4] Pereira,F.C.N. and Warren,D.H.D. : Definite Clause Grammar for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence, Vol.13(1980), pp231-278

[5] Chang,C.L. and Lee, R.C.T : Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York(1973)

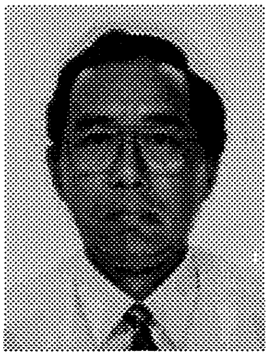
About the Authors



Hitoshi Inoue (井上　仁) was born in Fukuoka on March 17, 1964. He received the B.S. degree in 1986 in Mathematics and the M.S degree in 1988 in Information Systems from Kyushu University. Presently, he is an Assistant of Computation Center, Kyushu University. His research interests are natural language interfaces.



Setsuo Arikawa (有川　節夫) was born in Kagoshima on April 29, 1941. He received the B.S. degree in 1964, the M.S. degree in 1966 and the Dr.Sci. degree in 1969 all in Mathematics from Kyushu University. Presently, he is Professor of Research Institute of Fundamental Inforamtion Science, Kyushu University. His research interests include algorithmic learning theory, logic and inference in AI, and information retrieval systems.



Shun-ichi Takeya (武谷　峻一) was born in Fukuoka on September 2, 1945. He received the B.Eng. degree in 1968, the M.Eng. degree in 1970 and the Dr.Eng. degree in 1976 all in Communication Engineering from Kyushu University. Presently, he is an Associate Professor and the Vice Director of Computation Center, Kyushu University. His present interests include information retrieval systems and natural language interfaces.

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