A CONSIDERATION TO KNOWLEDGE REPRESENTATION-AN INFORMATION THEORETIC VIEW

Osuga, Setsuo
Institute of Interdisciplinary Research, Faculty of Engineering, The University of Tokyo

https://doi.org/10.5109/13363

出版情報：Bulletin of informatics and cybernetics. 21 (1/2), pp.121-135, 1984-03. Research Association of Statistical Sciences
バージョン：
権利関係：

Bulletin of Informatics and Cybernetics Vol. 21, No. 1～2, 1984

A CONSIDERATION TO KNOWLEDGE REPRESENTATION —AN INFORMATION THEORETIC VIEW

By

Setsuo OHSUGA*

Abstract

An information theoretic view is introduced into the discussion of knowledge representation. Information is defined to a description as the difference between the entropies the system contains before and after the description is given. Properties that should be attributed to knowledge representation are discussed taking this view into account.

Then, it is shown that an extention of first order predicate logic has these properties and is therefore suited for knowledge representation. A formula in this logic involves data structures in such a way that the data structures can be manipulated without having any influence upon this formula and, nevertheless, the formula evaluates correctly the data structures it refers to.

1. Introduction

Knowledge representation is the issue that involves some of the most fundamental problems of knowledge information processing. Nevertheless, theoretical foundation of this problem has not been established yet. In this paper we discuss an aspect of knowledge representation and derive some properties that should be attributed to it. Then we will show that a special extension of predicate logic has these properties. We introduce an information theoretic view in this discussion. Before going into details, we first present our view for the knowledge information processing.

(1) we think knowledge based system will be considerably useful if it will be able to support men in their creative works such as research studies, developments and designings. In particular, we consider that the design works in various fields, such as mechanical design, electrical and/or electronic design, material design, architectual design as well as software design will be important applications in near future.

Note that, in many application systems of knowledge engineering at present, for example in many medical consultation systems, one usually takes it for granted or, at least, allows that the conclusion derived from the system involves ambiguity and even inconsistency to some extent. It may be because nothing can be described exactly in such a system; not only the knowledge source but theoretical framework of the system are defined incompletely.

We can not, on the other hand, permit any ambiguity and inconsistency comming into the processes in CAD systems, because the results may be used directly in some

* Institute of Interdisciplinary Research, Faculty of Engineering, The University of Tokyo

machines to produce real objects. Then such systems must be capable of performing automatic check for consistency and reduction of ambiguity.

Consistency is a characteristic of knowledge base as the collection of all assertions (rules and facts). In other words, inconsistency arises as the interference between assertions.

Thence, we need a theory that governs knowledge base, with which we can analyse the behavior of the knowldge base and based on which the consistency and ambiguity are checked. As far as we know, however, most of present knowlede engineering systems lack the theoretical background in this sense. The only exception, we think, is to use predicate logic as knowldge representation language. In this case theory of predicate logic can support man to establish theoretical foundation of knowldge base. We use, therefore, the predicate logic as the theoretical framework of knowledge representation language design.

(2) Development is a very sophisticated task to create the model of object that satisfies the given requirements. Then, its support system must also be intelligent enough. In this paper, we discuss the technique to build the object model in the knowledge based system. We think that we need two different concepts to represent the model; (1) the model structures and (2) properties and relations of model components and also properties and behavior of the whole model. The former represents the structural relationship of the model and its components. It usually takes the form of data structure in the computers. The latter can be represented in terms of predicates. Ordinary predicate logic, however, is not always a convenient means for the purpose because it lacks the concept of data structure. We therefore, extend it to include that concept. We need some notions in doing it. Because the design work is the process to look for the structure that meets the given condition by the try-and-error operations, the tentative strucuture may be build and scrapped. During while, the requirement descriptions have to be remained unchanged. That is, the predicates have to be able to refer the structure to evaluate it correctly whatever the structure may be changed and, at the same time, it must be immutable by the changes of the structure. As such a predicate, we define a new logic named multi layer logic.

(3) Man learns many things from environments and updates his knowledge continuously. Then, human's knowledge has a nuance, that is, there is a delicate differences in meanings between expressions. In order that a knowledge based system, receiving this expression, accepts correctry the information involved in it, knowledge representation should also be able to represent the delicate differences. To discuss this issue, we define a quantitative measure to evaluate information contained in an expression. Our first objective of introducing such a measure was to make clear an aspect of knowledge representation. But it also gives us the important cue to find the relevant information in the knowledge base at the deductive operation.

In the following, we first define a measure for information contained in a logical formula in chapter 2. In chapter 3, we define multi layer logic, an expanded version of first order predicate logic to involve the concept of data structure. We show in chapter 4, that a formula of this new logic contains information that varies widely depending on the structure it refers to. Also we show in that chapter that this logic is suited as the knowledge representation for the development support systems.

2. Logical Formula and Its Information

Let the collection of all objects involved in the problem area to be dealt with by a knowledge based system be the world U, and the set of the predicate symbols be P. A term is either a constant or a variable defined in or over the world. (A function is also a term in the ordinary logic but it is represented in this paper in the form of logical predicate.) A predicate symbol in P followed by a finite set of terms in a sequence forms an atomic formula, or atom in short. Then, we can define the well formed formula, or formula in short, according to the ordinary definition. Let the formula be denoted

$$(Q_1x_1)(Q_2x_2)\cdots(Q_mx_m)M(x_1x_2\cdots x_m) \quad (\S1)$$

where x_i and Q_i, $i=1, 2, \cdots, m$, are the variables and the quantifiers (either $\forall$ or $\exists$) respectively. $M(x_1, x_2, \cdots, x_m)$ is a main body or a matrix of the formula formed of the finite set of atoms and connectives such as $\wedge$ (conjunction), $\vee$ (disjunction), $\sim$ (negation), $\rightarrow$ (implication) and $\leftrightarrow$ (equivalence). We consider only the closed formula (or the sentence) with no free variables, i.e., with every variable bounded. The world U over which every variable is defined involves various objects with the different attributes. Very often, objects that satisfy a given formula are restricted to a specific subset of U, which we call a sort. Ordinary first order logic represents a fact that an object x belongs to a sort d, by means of a predicate $D(x)$ meaning that x belongs to d. Then an expression "*every* (or *some*) element x in the sort d has the property $F(x)$" is represented by

$$(\forall x)[D(x)\rightarrow F(x)] \quad (\text{or} \quad (\exists x)[D(x)\wedge F(x)]) \quad (\S2)$$

There is a modification of the predicate logic called many sorted logic. It involves the concept of sort in the formula. Above expressions are represented, in this logic, as $(\forall x/d)F(x)$ and $(\exists x/d)F(x)$ respectively. A very simple example is "Every man is mortal". It is expressed in the first order one sorted logic as $(\forall x)[MAN(x)\rightarrow MORTAL(x)]$, while it is expressed in the many sorted logic as $(\forall x/man)MORTAL(x)$.

Let's consider a predicate F and let d be a finite set. For simplicity, we assume F being a single place predicate $F(x)$. It gives a description on an object. Or, in other word, $F(x)$ classifies all elements in the set d into two classes: those that satisfy $F(x)$ and those that do not. In the following, $F(x)$ and $\bar{F}(x)$ mean that "$F(x)$: True" and "$F(x)$: False" respectively for $x\in d$. Let's think of the state of d before and after the formula is given. In the prior state, whether $F(x)$ or $\bar{F}(x)$ is not clear for any x in d, while, in the posterior state, either $F(x)$ or $\bar{F}(x)$ is made clear for some or all elements in d.

Let $d=\{a_1, a_2, \cdots, a_N\}$ be a finite set. The state of d is defined as the conjunctions of either $F(a_i)$ or $\bar{F}(a_i)$ for every element a_i, $i=1, 2, \cdots, N$. Before the formula is given, the state of d includes all possibilities of $S_1: \bar{F}(a_1)\wedge\bar{F}(a_2)\wedge\cdots\bar{F}(a_N)$ through

$S_{2^N}: F(a_1) \wedge \cdots F(a_N)$. The set S_F^1 is defined as the collection of all possible prior states. That is,

$$S_F^1 = \{F\bar{a}_1\bar{a}_2 \cdots \bar{a}_N,\ F\bar{a}_1\bar{a}_2 \cdots,\ \bar{a}_{N-1}a_N,\ \cdots,\ Fa_1a_2 \cdots a_N\} \tag{§3}$$

where $F \cdots \bar{a}_i \cdots a_j \cdots$ denotes that $\bar{F}(a_i)$ and $F(a_j)$ for i th and j th elements. Let the cardinality (number of elements) of a set x be denoted $|x|$. Then $|d| = N$ and $|S_F^1| = 2^N$.

Elements of S_F^1 are mutually disjoint and exhaustive. Because we do not know in which state of S_F^1 d has been before some formula is given, we give the same state probability, 2^{-N}, to each element of S_F^1. Then we introduce the concept of entropy into S_F^1. Let it be I_{SF1}, then

$$I_{SF1} = -\sum_{SF1} \log 2^{-N} = N. \tag{§4}$$

Next, suppose a formula is given to the system. Only three forms in expression are possible by using many sorted logic; $F(a_i)(a \in d)$, $(\forall x/d)F(x)$ and $(\exists x/d)F(x)$. Any of them effects to confine the possible states of d to some subset of S_F^1 or to a specific state.

Consider, first, $F(a_i)$. Without loss of generality, we put $i=1$. To give $F(a_1)$ means to declare that, among S_F^1, states of the form $F\bar{a} \cdots$ can never arise. Since, however, it says nothing on the elements $a_2, \cdots, a_N$, the set of possible posterior states, which we denote S_F^2, is,

$$S_F^2 = \{Fa_1\bar{a}_2 \cdots \bar{a}_N,\ \cdots,\ Fa_1a_2 \cdots a_N\}.$$

Here we introduce still a new notation for simplicity. Let the part of suffix that is fixed by the given formula be enclosed by the parenthesis and every combination be made to the remaining part. Then, S_F^1 and S_F^2 are denoted respectively,

$$S_F^1 = \{Fa_1 \cdots a_N\}, \qquad S_F^2 = \{F(a_1)a_2 \cdots a_N\}. \tag{§5}$$

The cardinality and entropy of S_F^2 are

$$|S_F^2| = 2^{N-1} \tag{§6}$$

$$I_{SF2} = N-1. \tag{§7}$$

The difference of I_{SF1} and I_{SF2} is the volume of information with respect to the predicate symbol F the forinula $F(a_1)$ brought in. We denote it K. Then

$$K = I_{SF1} - I_{SF2} = N - (N-1) = 1.$$

Next, $(\forall x/d)F(x)$ means that, for every a_i in d, $F(a_i)$, then

$$S_F^2 = \{F(a_1 \cdots a_N)\} \tag{§8}$$

$$|S_F^2| = 1 \tag{§9}$$

$$I_{SF2} = 0 \tag{§10}$$

$$K = I_{SF1} - I_{SF2} = N. \tag{§11}$$

Because $(\exists x/d)F(x)$ allows every element in S_F^1 except $F\bar{a}_1 \cdots \bar{a}_N$,

$$S_F^2 = S_F^1 - \{F(\bar{a}_1 \cdots \bar{a}_N)\} \tag{§12}$$

$$|S_F^2| = 2^N - 1 \tag{§13}$$

$$I_{SF2} = \log(2^N - 1) \tag{§14}$$

$$K = N - \log(2^N - 1)\,. \tag{§15}$$

Information, brought in by the formula in this case, is very small.

There are another set of possible formulas that use $\sim F$ instead of F. There are, however, no difference in the discussion of this case from that and is omitted.

The above discussion is easily extended to the case of n place predicate with n greater than one. For example, consider $(\exists x/d_1)(\forall y/d_2)F(xy)$, where $|d_1| = M$, $|d_2| = N$. Information is obtained to this formula by enumerating every possible combination of elements in the space of $d_1 \times d_2$ that satisfies the formula. The cardinality of such a set of elements is obtained by simple computation as $|S_F^2| = 2^{M \times N} - (2^N - 1)^M$. Then information is obtained as the difference of the prior and posterior entropies, as

$$K = M \times N - \log[2^{M \times N} - (2^N - 1)^M]\,. \tag{§16}$$

3. Expansion of Predicate Logic—Multi Layer Logic

3.1 Knowledge representation usable for model building

We note from this discussion that, (1) many sorted logic (or predicate logic, in general) allows very incomplete expressions (those with large entropies) and (2) the distance between information in allowable expressions in the syntax of many sorted logic is fairly large.

As mentioned before our objective is to realize the development support system. Here we consider that development is a process to give information successively to build and refine the object model. The property (1) above of the predicate logic is favorable for man in man-machine interaction, while, the property (2) is not because it requires man being engaged in the creative work to give large volume information in a step. He may prefer to give it successively by a small amount. Many sorted logic is not always good means because of the property (2) above and we wish to have a means for knowledge representation that allows one to give information to the model by a small amount in each interaction. While it is undesirable if this new means would be so complicated that theoretical framework of predicate logic is destroyed.

We have also discussed that the object model is represented by means of structures and predicates. Given a set of requirements, man works to find the model structure to meet them. As the ordinary process he makes a tentative structure and evalutes it whether it meets the given requirements. If not, he shoud modify the structure.

During this repetitive process, the structure varies widly while the requirement description should remain unchanged. This fact requires that the set of requirements has to be applied to the structure to evaluate it correctly whatever it may be changed, while the requirements description must be insensible to the change in the structure. We define, as the knowledge representation to meet this condition, an extended version of many sorted logic named multi layer logic. We begin with the discussion on involving structures in the predicate logic.

3.2 Many sorted logic and structure involved in it

The basic difference between first order one sorted logic and many sorted logic lies in the fact that the former does not involve any concept of structure while the latter involves the concept of structures composed of set-element relations. This structure will naturally induces the set-subset relation among objects when the logic applies to some real word. It is possible, then, by using this mechanism, to define multi stage relations of the form $\cdots x \subset y \subset z \cdots$ in the world and represent it in the compututer systems as a hierarchical data structure. This structure can be used conveniently to represent a concept structure to form a model of the whole world. However, it is not always powerful enough for representing the object model structure and we need another type of hierachical structure.

3.3 An extension of many sorted logic—multi layer logic

Let $d=\{a_2, a_2, \cdots, a_N\}$ be a finite set and $b_i=\{a_{i1}, a_{i2}, \cdots, a_{im_i}\}$, $i=1, 2\cdots, k$, be its subsets. Suppose we have a machanism to link each subset to some another object. For example, let a_i, $i=1, 2, \cdots, N$, represents a line defined in a three dimensional space.

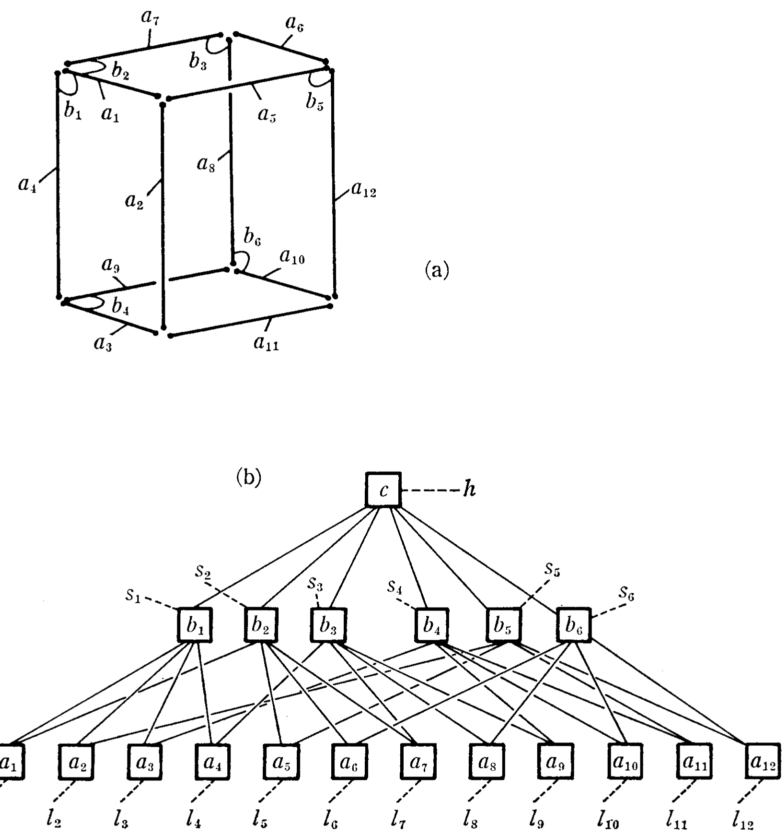


Fig. 1 Representation of a three dimensional structure

Sometimes, we use such a convention as to represent a surface by the set of its edge lines. That is each b_j, $j=1, 2, \cdots, m$, is interpreted as representing a surface, s_j. Similarly, a set $c=\{b_1, b_2, \cdots, b_k\}$ represents an instance of polyhedron h. Fig. 1 shows such an example. We assume here is a mathematical reration $a_i \in s_j \in h$, and accordingly, $a_i \in b_j \in c$. This is another type of relation from that induced by many sorted logic. This relation can be defined mathematically by introducing the concept of power set.

When the set d is given, we denote by $*d$ the set composed of all subsets of d but excluding the empty set. For example, when $d=\{1, 2, 3\}$, $*d=\{(1), (2), (3), (1, 2), (1, 3), (2, 3), (1, 2, 3)\}$.

Since $*d$ is itself a set, $*(*d)$ can also be defined in the same way, denoted by $*^2d$. In general, $*^nd$ can be defined recursively as $*^nd=*(*^{n-1}d)$.

Let $c \in *^2d$. Then there is some $a_i \in d$ and $b_j \in *d$ in relation $c \ni b_j \ni a_i$ with respect to c. If d is the set of lines, c can be a polyhedron. Thus, in general, $*^nd$ represent a set of all possible structures, composed of and being n-levels higher than the component set d, in the sense the polyhedron is the structure composed of and two levels higher than the set of lines.

Suppose that a description is given on some component(s) of this structure, such as "The length of some line in the surface $s(=b_j)$ is $5(cm)$". Though this description does not contain as much information as to identify exactly which line, but it contain some non-zero amount of information. It is natural to represent it in the style of many sorted logic.,

$$(\exists x/b_j)(LENGTH\ x\ 5)$$

where $(LENGTH\ u\ v)$ means that the length of u is v. Similarly, $(\forall y/c)(AREA\ y\ 20)$ means that the area of every surface of the polygon $h=c$ is $20^{(cm^2)}$ where $(AREA\ u\ v)$ means that the area of u is v. Then it is also possible to put these formulas in the same formula.

$$(\forall y/c)(\exists y/b)[(AREA\ y\ 20)\wedge(LENGTH\ x\ 5)]. \qquad (\S 17)$$

Though, in this formula, both y and b represent some surfaces, there is no direct relation between the assignments of surfaces to them. But it must be possible to make them identical if we want to represent the description, "For every surface in the polyhedron, the area of the surface is $20^{(cm^2)}$ and there is, in this surface, a line of length $5^{(cm)}$". Direct translation of this sentence generates

$$(\forall y/c)(\exists x/y)[(AREA\ y\ 20)\wedge(LENGTH\ x\ 5)] \qquad (\S 18)$$

which is the same with (§17) with the exception that b_j is replaced by the variable y appearing in the top of the prefix. This requires that many sorted logic should be extended so that the sort is replaced by an arbitrbry set which can itself be a variable. We call the extention multi layer logic. To make clear the distinction between the fixed sets and variable sets in the expression, we attach the symbol $\sharp$ to the former. For example, (§17), (§18) are witten

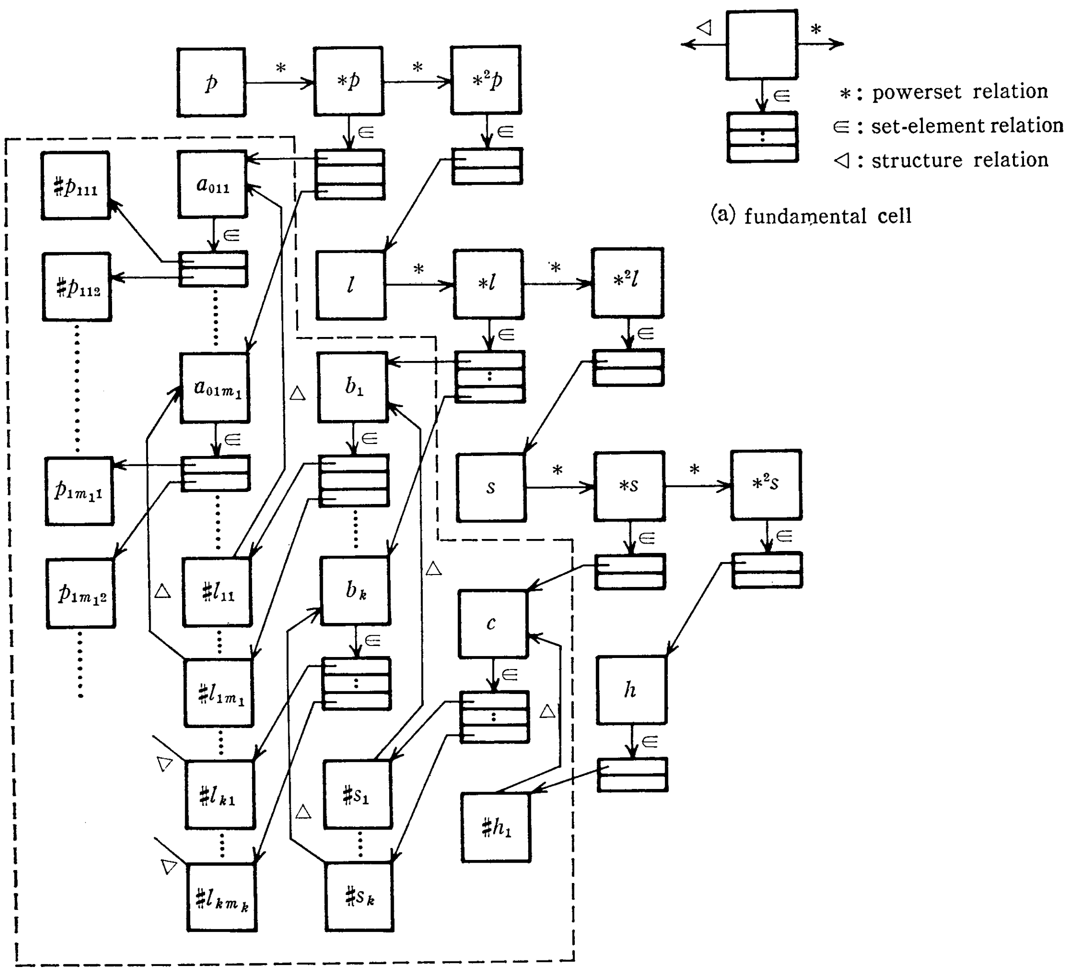
$$(\forall y/\sharp c)(\exists x/\sharp b)[(AREA\ y\ 20)\wedge(LENGTH\ x\ 5)] \qquad (\S 17')$$

$$(\forall y/\sharp c)(\exists x/y)[(AREA\ y\ 20)\wedge(LENGTH\ x\ 5)]. \qquad (\S 18')$$

Note, however, that the symbol $\#$ is not an essential one to define multi layer logic. It is introduced only for convenience in explanation.

As was the fixed sort in many sorted logic a mechanism to define a structure based on the set-subset relation in the real world, the variable set in multi layer logic is the mechanism to define another structure based on the sequence of set-element relations, that is, $\cdots x \in y \in z \cdots$. Thus multi layer logic allows to specify two kinds of structures among objects in the real world; $u \subset v \subset w \cdots$ and $x \in y \in z \cdots$. We identify them as Type 1, or $T1$ in short, structure and Type 2, or $T2$, structure.

Note here that $T1$ structure is formed very naturally as the result of classifying objects in the real world while $T2$ structure is rather an artificial structure. It is formed by connecting an instance object to a set of the other objects like connecting s_j to the set b_j in the previous example. This connection does never imply the inherent relationship between these objects but involves a view of human user in representing objects



(b)

Fig. 2 An example of object structure in the world composed of the other objects. Lets the connection be denoted by the symbol $\rhd$, shown as $s_j \rhd b_j$. Then the surface and line relation $s_j \ni a_i$ is, in fact, implemented as the compound relations $s_j \rhd b_j$ and $b_j \ni a_i$.

When we intend to use multi layer logic, it is convenient to represent these structures in the real world by the data structure. Such a data structure is shown in Fig. 2. Every object in the real world is given a fixed length cell in this data structure with the set of pointers to the other nodes to show the specified relations between this node and the pointed nodes (objects). Then, the formula in multi layer logic is connected to the structure. Each (Qx/d) in the prefix of the formula, say (§18), tells that x and d is in the reration $x \in d$. Then, by referring to the node representing d in the data structure, the scope of x is easily found. When d is an instance object, the formula is specified to refer the $T2$ structure, then we should trace the data structure first by the pointer labelled $\rhd$ from d and then find the objects x that is in the relation (Qx/d). On the other hand, when d denotes a set, we can directly find x such as (Qx/d). Thus the route selection is necessary depending on the content of the given formula. To facilitate the selection we introduce the special symbol. For example, (§18′) is written as

$$(\forall y /\!/ \sharp c)(\exists x /\!/ y)[(AREA\ y\ 20) \wedge (LENGTH\ x\ 5)] \qquad (\S 18'')$$

where $(Qx /\!/ y)$ tells that the type 2 structure is to be refered. Note that, similar to the symbol $\sharp$, the use of dould slant is not essential for multi layer logic but only for convenience for computer processing. Also note that we did not impose any condition to define $T2$-structure other than the finiteness of the base set d. The elements of d can be any object and are not necessarily be required the homogeneity because, when we disscuss the structure composed of the base set, the detail of each element is completely hidden. For example, it can be the set of different machine elements to compose the machine systems or the set of different atoms to compose the chemical compound or so.

3.4 Problem of synthesis

There are two different types of problems concerning to dealing with the object model. The first is an analytic problem where the object structure is given and properties, behavior etc. that are to be decided depending on the structure are to be obtained. The second is the synthetic problem where the base set and requirement on properties, behavior etc. are given and the structure that satisfies these requirements is to be organized. For example, the analytic problem is, given the polyhedron as shown in Fig. 1, to obtain its volume while, the synthetic problem is, given the set of lines, to organize the polyhedron with specified volume. Since such a structure is an element of $*^2d$ where d is a set of lines, the requirement to this synthetic problem is expressed,

$$(\exists z /\!/ *^2\sharp d)(VLOUME\ z\ 100)?$$

The more general from of the synthetic operation involves the conditions on the intermediate structure, and is represented as,

$$(\exists z^n /\!/ *^n\sharp d)(\forall z^{n-1} /\!/ z^n) \cdots (\forall z^1 /\!/ z^2)[M(z^n, z^{n-1}, \cdots, z^1)]\,. \qquad (\S 19)$$

As an example, the description on how the polyhedron can be organized from the given set of points (instead of lines) in the three dimensional space is given. Let the set of points be $d=\{p_1, \cdots, p_N\}$. Then

$$(\forall z /\!/ *^3\# d)(\forall y /\!/ z)(\forall x /\!/ y)[(COUNT\ x\ 2)\wedge(ELEM\text{-}CYCLE\ y)\wedge(EULAR\text{-}EQ\ z) \rightarrow (POLYHEDRON\ z)]\,. \quad (\S 20)$$

When the set of lines was used as the base set, a polyhedron was an element of $*^2d$. When the base set is the set of points, the every polyhedron is an element in $*^3d$ that satisfies a set of conditions to be a polyhedron. Then (§20) gives these conditions. Here, each predicate has the following meaning:

$(COUNT\ x\ n)$: The cardinality of (the set) x is n.
$(ELEM\text{-}CYCLE\ y)$: (The set) y forms an elementary cycle.
$(EULAR\text{-}EQ\ z)$: (The structure) z satisfies Eular equation.
$(POLYHEDRON\ z)$: (The structure) z is a polyhedron.

$(COUNT\ x\ 2)$ shows that x is a pair of points (the set of just two elements) and represents a line, $(ELEM\text{-}CYCLE\ y)$ shows that a set y (of lines) should form a closed cycle to represent a surface. Here a convention to represent a surface by its edge lines is adopted. $(EULAR\text{-}EQ\ z)$ requires that z satisfies Eular equation, which gives the condition of z being a polyhedron. As is well known a polyhedron with no hole in it satisfies this condition.

Thus multi layer logic allows to define $T2$-structure in the world. This is indispensable to represent an object model structure in the form of data strucuture. The noticeable feature of formulas of multi layer logic is in its prefix that relates with the structure in the world. Because the formula is not necessarily bound to a specific structure, it can be insensible to the changes of the data structure it relates to. This is a quite desirable property when we intend to express the requirements, particularly, in synthetic problems.

In order to make the definition of multi layer logic exact, however, we need an additional notion. One of the typical forms of multi layer logic is

$$(\exists x^n /\!/ *^n\# d)(\forall x^{n-1} /\!/ x^n) \cdots (Q_1 x^1 /\!/ x^2) M(x^n \cdots x^1)\,. \quad (\S 21)$$

For simplicity, let's consider by the example,

$$(\exists x^2 /\!/ *\# d)(\forall x^1 /\!/ x^2) F(x^1)? \quad (\S 22)$$

This is satisfied by such x^2 as: (1) $x^2 \subset d$ and (2) for every element x^1 of x^2, i.e. $x^1 \in x^2$, $F(x^1)$. Note here that $x^2 \in *d$ is equivalent to $x^2 \subset d$.

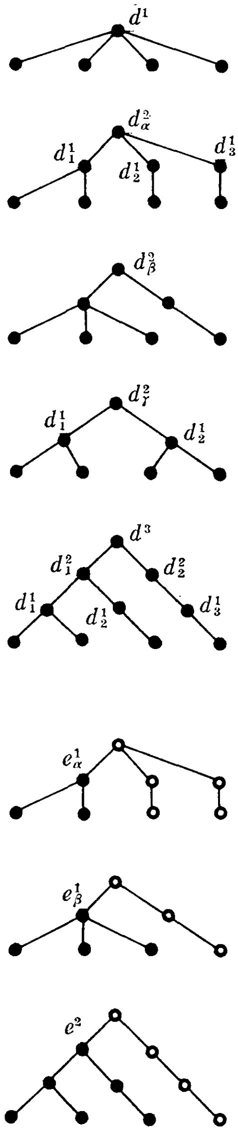
Suppose such an x^2 is found. But it is not a unique solution because there are many values for x. For example, every y such that $y \subset x^2$ satisfies these conditions. In order to obtain unique solution, we add third condition. (3) Among all x^2's that satisfy the condition (1) and (2), $\hat{x}^2$ such that $\hat{x}^2 \supset x^2$ is the only solution. This discussion holds true in the general case of (§21).

Now compare (§22) with the formula of many sorted logic, say, $(\exists x/\# d)F(x)$. Both obtain x that satisfy $F(x)$. But the former requires to obtain the set of all x's that satisfy $F(x)$ while tha latter is true when at least one such x is found.

Next consider the similar formula to (§22).

$$(\exists x^2 /\!/ {}^*\sharp d)(\forall x^1 /\!/ x^2)[G(x^2) \wedge F(x^1)]? \qquad (§23)$$

This requires first to obtain the set $x^2 \subset d$ such that every element x^1 of x^2 satisfies $F(x^1)$, and then apply $G(x^2)$. If, for example, $G(x^2)$ is a predicate that is related to some statistical program routine, then (§23) will derive the statistical value of a collection of data which satisfy some given condition $F(x^1)$ among all data.



		(b)	(c)
[A]	(i)	$(\exists x /\!/ d^1)F(x)$	4--log 15=0.008
		$(\forall x /\!/ d^1)F(x)$	4
	(ii)	$(\exists y /\!/ d_\alpha^2)(\forall x /\!/ y)F(x)$	4−log 13=0.300
		$(\forall y /\!/ d_\alpha^2)(\exists x /\!/ y)F(x)$	4−log 3=2.415
	(iii)	$(\exists y /\!/ d_\beta^2)(\forall x /\!/ y)F(x)$	4−log 9=0.830
		$(\forall y /\!/ d_\beta^2)(\exists x /\!/ y)F(x)$	4−log 7=1.193
	(iv)	$(\exists y /\!/ d_\gamma^2)(\forall x /\!/ y)F(x)$	4−log 7=1.193
		$(\forall y /\!/ d_\gamma^2)(\exists x /\!/ y)F(x)$	4−log 9=0.830
	(v)	$(\exists z /\!/ d^3)(\forall y /\!/ z)(\exists x /\!/ y)F(x)$	4−log 11=0.541
		$(\forall z /\!/ d^3)(\exists y /\!/ z)(\forall x /\!/ y)F(x)$	4−log 5=1.678
[B]	(i)	$(\exists x /\!/ e_\alpha^1)F(x)$	4−log 12=0.415
		$(\forall x /\!/ e_\alpha^1)F(x)$	2
	(ii)	$(\exists x /\!/ e_\beta^1)F(x)$	4−log 14=0.193
		$(\exists x /\!/ e^2)F(x)$	3
	(iii)	$(\exists y /\!/ e^2)(\forall x /\!/ y)F(x)$	4−log 10=0.678
		$(\forall y /\!/ e^2)(\exists x /\!/ y)F(x)$	4−log 6=1.415

(a) (b) (c)

Fig. 3 Information contained in a formula

4. Information of Multi Layer Logic

A formula of multi layer logic shows various different information depending on the structure it refers to. We show it by an example. Let $d=\{a_1, a_2, a_3, a_4\}$ be a fiite base set. Various $T2$-structure composable of this set are shown in Fig. 3(a). Since each node in these structures represents a real object, all nodes that are related to the real objects in the same category, for example, surfaces, should be located at the same level.

Fig. 3(b) shows the possible formulas in relation with the structure and Fig. 3(c) is the information that the formula contains. This is easily obtainable. We show it by an example for the case $(\exists y /\!/ d_\beta^2)(\forall x /\!/ y)F(x)$ of structure (iii).

The set of possible states under the formula is

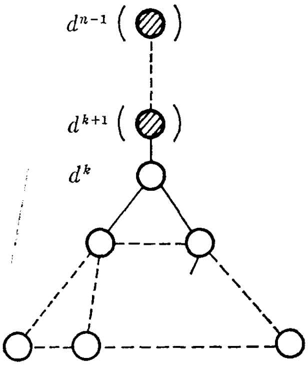
$$S_F^2=\{F(a_1a_2a_3)a_4,\ Fa_1a_2a_3(a_4)\}\ .$$

Then, $|S_F^2|=9$ and $K=\log|S_F^1|-\log|S_F^2|=\log 4-\log 9=0.830$. This simple example of Fig. 3 shows that a formula of multi layer logic reveals various different information depending on the structure it relates to. The maximum information is N when $|d|=N$ while the minimum is $N-\log(2^N-1)$ as before. But there are many intermediate values between these extremes, and, as N increases, the relative distance (with respect to the max. value) of information between ajacent possible expressions decreases.

It seems that the expressions in Fig. 3(b) are different to each other. But it is shown easily that these expressions are, in fact, identical under the transformation operation between the equivalent expressions. There are structures with the different heights in Fig. 3(a). The maximum height of the structure is $N-1$ ((V) in Fig. 3) when the base set involves N elements. It is possible, then, to normalize all structures to the one with the max. height by adding, if necessary, node(s) over the top node as shown in Fig. 4. Then the expression is also normalized to

$$(Q_{n-1}x^{n-1} /\!/ \# d^{n-1})(Q_{n-2}x^{n-2} /\!/ x^{n-1}) \cdots (Q_1x^1 /\!/ x^2)F(x^1) \qquad (\S 24)$$

where Q_i, $i=1, 2, \cdots, n-1$, denotes either $\forall$ or $\exists$. If $d^{n-1}=\{d^{n-2}\}$, that is, if the top node is a set of singleton, x^{n-1} is restricted to d^{n-2} and (§24) can be changed to the equivalent formula with the structure of which the height is one level lower,



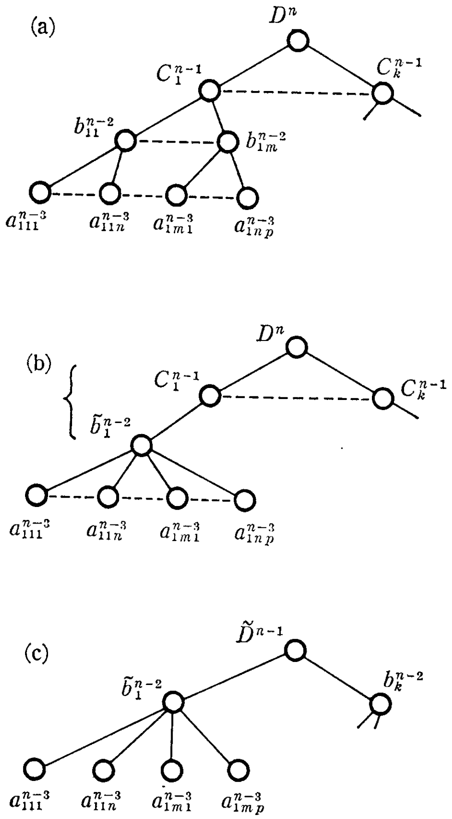
$$(Q_{n-1}x^{n-1} /\!/ d^{n-1})(Q_{n-2}x^{n-2} /\!/ p^{n-1})(Q_kx^k /\!/ x^{k+1}) \cdots (Q_1x^1 /\!/ x^2)F(x^1)$$

$$d^{n-1}=\{d^{n-1}\},\ Q_n=\forall \text{ or } \exists$$

$$=(Q_{n-2}x^{n-2} /\!/ d^{n-2}) \cdots (Q_kx^k /\!/ x^{k+1}) \cdots (Q_1x^1 /\!/ x^2)F(x^1)$$

$$=\cdots=(Q^q x^k /\!/ d^k) \cdots (Q_1x^1 /\!/ x^2)F(x^1)$$

Fig. 4 Equivalent expressions (1)



$$(Qz/\!/D^n)(\exists y/\!/z)(\exists x/\!/y)F(x)$$
$$D^n=\{c_1^{n-1},\cdots,c_k^{n-1}\}$$
$$\parallel$$
$$(Qz/\!/D^n)(\exists y^1/\!/z)(\exists x/\!/y^1)F(x)$$
$$D^n=\{c_1^{n-1},\cdots,c_k^{n-1}\},$$
$$c_i^{n-1}=\{\tilde{b}_i^{n-2}\},\quad i=1,2,\cdots k,$$
$$\parallel$$
$$(Qy^1/\!/\tilde{D}^{n-1})(\exists x/\!/y^1)F(x)$$
$$\tilde{D}^{n-1}=\{b_1^{n-2},\cdots,b_k^{n-2}\},$$

Fig. 5 Equivalent expressions (2)

$$(Q_{n-2}Z^{n-2}/\!/\#d^{n-2})\cdots(Q_1Z^1/\!/Z^2)F(Z^1). \tag{§25}$$

This operation is repeated to come back to the original structure and the expression.

Next, suppose that the same quantifiers appeared in succession in the prefix of a formula along the structure it refers to as shown in Fig. 5(a). The structure, then, can be changed as shown in Fig. 5(b) without changing the meaning of the formula. Furthermore, both the formula and the structure are changed as shown in Fig. 5(c) preserving the equivalence.

Hence, we can consider only formulas such that the quantifiers change alternatively in the prefix. Then, given the base set, there are only two different classes of formulas $(\exists x^{n-1}/\!/d^{n-1})(\forall x^{n-2}/\!/x^{nn-1})\cdots(Q^1x^1/\!/x^2)F(x^1)$ and $(\forall x^{n-1}/\!/d^{n-1})(\exists x^{n-2}/\!/x^{n-1})\cdots(Q^1x^1/\!/x^2)F(x^1)$. All other froms are equivalent either of them. This is the same situation with the ordinary logic has also two different expressions: $(\exists x)F(x)$ and $(\forall x)F(x)$.

Next consider the case of synthetic processes. When $|d|=N$, $|{}^*d|=2^N-1$. In general, $|{}^{*n}d|=2^{|{}^{*n-1}d|}-1$. When N is large, $|{}^{*n}d|$ can be approximated by $2^{|{}^{*n-1}d|}$. Suppose a formula defined on the set ${}^{*n}d$ selects M structures out of $2^{|{}^{*n-1}d|}$ posssible structures. Then, prior and posterior entropies of the system are $2^{2|{}^{*n-1}d|}$ and 2^M respectively. Information is $K\doteqdot 2^{|{}^{*n-1}d|}-M$.

In general, $2^{|*^{n-1}d|}$ is very large when d is large. Then the large volume of information is necessary to select one or a few structures out of possible structures. This is one reason of the synthetic problem being ordinarily very difficult.

The effective way of synthesizing objects is to decrease possibilities as early stage as possible. In many cases, it is realized by giving the conditions on the intermediate structures. COUNT and ELEM-CYCLE in the formula ($20) were the examples.

Then, how effective is it to give the constraints on the intermediate structure ? We estimate it by the example of COUNT above.

First, we see how much is the information contained in $(\exists x/\!/{*d})$ (COUNTx2). The number of elements (sets of points) before and after the formula is given are 2^N and ${}_NC_2$. Then $K=2^N-{}_NC_2$. Next the number of elements in $*^3d$ is $2^{2^{2^N}}$. When $(\exists x/\!/{*d})$ (COUNTx2) is used before to generate $*^3d$, the number decreases to $2^{2^{{}_NC_2}}$. The difference of entropies between these cases is $K'=2^{2^{2^N}}-2^{2^{{}_NC_2}}$. It is far much greater than $K=2^N-{}_NC_2$, thus the information at the intermediate stage is considerably amplified.

5. Conclusion

We have discussed, in this paper, some problems involved in the issue of knowledge representation. We introduced the information theoretic view in this discussion.

In particular, we have discussed the conditions to be imposed on knowledge representation for the knowledge based system being able to support human user in his creative works. We have difined a modification of first order predicate logic and named it multi layer logic. The characteristic of the logic are as follows: (1) it allows to use data structure together with the logic to represent the real world in the computer, (2) the logical formula has relation(s) with the data structure(s) in the real world to describe something on the structure, (3) the structure can be defined and modified independent of any logical formula referring to it and the logical formula is insensible to any change in the structure, (4) the same logical formula can have various volumes of information depending on the structure it is referring to.

What we have discussed in this paper is only a part of disucussions necessary to make clear the concept of knowledge representation. We think we need more discussions on such issues as the proof procedure by multi layer logic, information theoretic view of the logical inference, quantitative evaluation of ambiguity and reduncy involved and representable within the framework of multi layer logic and so on.

Acknowledgements

I wish to thank Mrs. Michiko Shimamura for her excellent support for making this paper.

References

[1] BARR, A. and FEIGENBAUM, E. A.: The Handbook of Artificial Intelligence, Vol. 1, Chap. 3, Wiliam Kaufmann Inc. (1981), 141-222.

[2] BRACHMAN, R. J. and SMITH, B. C.: *Special Issue on Knowledge Representation*, SIGART

Newsletter, 70, (1980), 1-138.

[3] CHANG, C. L. and LEE, R. C. T.: Symbolic Logic and Mechanical Theorem Proving, Academic Press (1973).

[4] ENDERTON, M. B.: A Mathematical Introduction to Logic, Academic Press (1979).

[5] OHSUGA, S.: *Perspectives on New Commputer Systems of the Next Generation—A proposal for Knowledge-Based Systems,* J. of Information Processing, 3, No. 3, (1980), 171-185.

[6] OHSUGA, S.: *A New Method of Model Description—Use of Knowledge Base and Inference—,* in CAD System Framework (ed. Bo, K. and Lillehagen, F. M.), North-Holland, (1983), 285-312.

[7] OHSUGA, S.: *Predicate Logic Involving Data Structure As A Knowledge Representation Language,* Proc. '83 Eighth International Joint Conf. on Artificial Intelligence, (1983), 391-394.

[8] OHSUGA, S.: *Knowledge Based Man-Machine Systems,* Automatica, Vol. 19, No. 6, (1983), pp. 685-691.

[9] ROBINSON, J. A.: *A Machine Oriented Logic Based on the Resolution Principle,* J. ACM, 12, (1965).

[10] WATANABE, S.: Knowing and Guessing—A Formal and Quantitative Study, John Wiley and Sons Inc. (1969).

Received November 7, 1983