数理解析研究所講究録
396 巻 1980 年 104-130

INFORMATION SPACE MODEL

Yuzuru Tanaka

Graduate School of Information Engineering
Hokkaido University
Sapporo, 060 Japan

The importance of the semantic theories of databases has come to be recognized in various problems of databases. This paper proposes an infosemantic framework based on the relational model that has been an infological framework of database theories. In a relational database, semantic relationships are partly embedded in attribute names and partly embodied by intrarelational semantics. However, most of them are hidden in interrelational relationships that are not explicitly specified by a relational schema. There are two kinds of interrelational structures, analytic ones and synthetic ones. Among them, the synthetic structures play especially important roles in semantic problems. However, there are no proper theoretical basis to deal with these structures. The information space model (R, M) gives an infosemantic framework of this problem, where synthetic structures are formalized as morphisms between relations. The paper gives detail formalization of this model and examples of its applications.

1. INTRODUCTION

Recent studies on databases indicates the necessity of a new theory about formal semantics of databases. Although the lack of formal semantics of databases has come to be noticed through the studies on relational databases, it is not only a problem of this special model but also a more general problem involving all kinds of data models. This problem unfortunately attracted very little attention before because of its difficulties.

Various approaches are possible to cope with this problem, however, we will choose the relational model as the basis of our approach to a new semantic model since the relational model has contributed a lot for these ten years to the development of database theories and we should not neglect this fact.

In a relational database, semantics of information is partly embedded in the names of attributes and partly in the intrarelational relatioships. However, most part of the semantics is hidden in the interrelational relationships that are not explicitly defined as a part of schema description. It may sound reasonable that someone says the interrelational semantics is described by relation names. However, this is the most common misunderstanding of relational semantics. While we can infer the semantic relationship between two relations from their names, it is absolutely impossible for the computer system that manages these relations to do the same inference.

To cope with the interrelational semantics of relational databases, we propose the information space model (R, M_0), where R is a set of first normal form relations of an object database, and M_0 is a set of elementary morphisms among elements in R. An interrelational relationship is described by a morphism between relations that is either elementary or derived from elementary ones by composition. Since the number of elementary morphisms is proved to be always finite, there always exists a finite description (R, M_0) for any database.

The elementariness of a morphism is clearly defined. For each morphism σ in M_0, we define a label $l(\sigma)$. A set $L(M_0)$ denotes $\{l(\sigma)|\sigma \in M_0\}$. A semantic attribute is a concatenation of an attribute and a list of labels, i.e., $Al(\sigma_1)l(\sigma_2) \ldots l(\sigma_n)$. Let $w(\sigma)$ denote the semantics of σ that may be informally interpreted as an English noun representing the relationship denoted by σ. Then the semantic attribute $Al(\sigma_1)l(\sigma_2) \ldots l(\sigma_n)$ may be informally interpreted as an English noun phrase A $prep_1$ $w(\sigma_1)$ $prep_2$ $w(\sigma_2)$ ... $prep_n$ $w(\sigma_n)$, where $prep_i$ is one of the following prepositions; 'of', 'in', 'at', 'on', and 'by' etc.

For an information space schema (R, M_0), a set M_0 generates a set of semantic attributes $\{A\rho|\ \rho$ is a finite sequence of elements in $L(M_0)$, i.e., $\rho \in L^*(M_0)\}$. This set is denoted by Ω^*, where Ω denotes a set of all the attributes appearing in some relation in R. For each σ in M_0, we can define a morphism $\hat{\ }\sigma$ between semantic attributes such that

$$\hat{\ }\sigma : A\rho \longmapsto Al(\sigma)\rho \quad \text{for } A \in \Omega \text{ and } \rho \in L^*(M_0).$$

A set of morphisms $\{\hat{\ }\sigma|\sigma \in M_0\}$ is denoted by $\hat{\ }M_0$. While (R, M_0) is a finite category, the category $(R^*, \hat{\ }M_0)$ is infinite. It should be noticed here that an infinitely large space $(R^*, \hat{\ }M_0)$ can be defined by a finite description (R, M_0).

In the following sections, after informal introduction of information space model, its formal semantics is formalized. Recursive morphisms and their relationships to schema design are detailed. And finally, denotational semantics of query language vocabulary is explained.

2. INTERRELATIONAL SEMANTIC STRUCTURE

2.1. Analytic Structures and Synthetic Structures

Interrelational semantic structures of a relational database are classified into two categories, i.e., analytic structures and synthetic structures.

Fig.1 (a) shows an example relation in the first normal form. This can be decomposed into two relations shown in (b) because of the existence of a functional dependency /department/$\rightarrow$/floor/. This decomposition process defines an interrelational semantic structure between R1 and R2 that reflects

R	employee	department	floor
	J. Smith	A	2
	K. Jones	A	2
	F. Brown	B	3
	:	:	:

(a) an original relation

R1	employee	department
	J. Smith	A
	K. Jones	A
	F. Brown	B
	:	:

R2	department	floor
	A	2
	B	3
	:	:

(b) two relations obtained by the decomposition of (a).

Fig. 1. An example of an analytic interrelational relationship.

the dependency structure they had in (a) before the decomposition. This kind of interrelational relationships is determined by the analysis of the intra-relational dependency structures of the original relation, and hence it is called an analytic structure. The original relation is a so called universal relation of R1 and R2.

However, we can not always assume the existence of a universal relation. Fig.2 (a) shows the instances of two relations for which there exists no universal relation. They are projections of a relation with a lot of null values (Fig.2 (b)). Fig.3 (a) shows an instance of a relation for which we can define a relation with infinitely many attributes (Fig.3 (b)). In these two examples, interrelational relationships are defined by something other than analytic dependency structures. Since their semantics is defined by the way of synthesizing an integrated view of information from original relations, we call such a structure a synthetic structure. Especially, the relationship in Fig.3 (a) is called a recursive synthetic structure. Recursive synthetic structures form a very interesting and important class of synthetic structures.

While there may exist more than one synthetic relationships between two relations, the analytic relationship between them is always unique if any. While analytic structures concern the decomposition of a first normal form universal relation, synthetic structures concern the overall semantic structures of a set of constituent first normal form relations. This paper deals with the synthetic structures. Our approach to analytic structures is detailed in [TANA77] and [TANA79].

2.2. Necessity of Denotational Interrelational Semantics

We show examples of three kinds of problems concerning the necessity of denotational description of interrelational semantics.

The first problem concerns the isomorphic relationship between a query language and a natural language. In Fig.4 (a), we show four example queries to a database in Fig.3 (a) written in both English and a SEQUEL like language [CHAM76]. While the representations of these queries in English are isomorphic, their representations in a SEQUEL like language have different forms. If we view this database as an infinite relation in Fig.3 (b) with an extended set of attributes then the representations of these queries in this query language become isomorphic as in Fig.4 (b). In these example queries, there appears two extended attributes, i.e., /name of the parent/ and /birth date of the parent/. They have the phrase "of the parent" in common. In English, these two appearances of "of the parent" have the same meaning. Obviously, the phrase "of the parent" is a kind of synthetic structures in this database. What is the formal semantics of "of the parent" in this database?

The second problem concerns the formal description of a subspace that is semantically meaningful in a real world of information. In the database in

novel	author
The adventure of Tom Sawyer	Mark Twain
Crime and Punishment	Feodor Dostoyevsky
For Whom the Bell Tolls	Ernest Hemingway
Gone with the Wind	Margaret Mitchell

character	novel
Porfiry	Crime and Punishment
Ishmael	Moby-Dick
Robert Jordan	For Whom the Bell Tolls
Philip Carey	Of Human Bondage
Pilar	For Whom the Bell Tolls

(a) two relations with a synthetic interrelational relationship between them

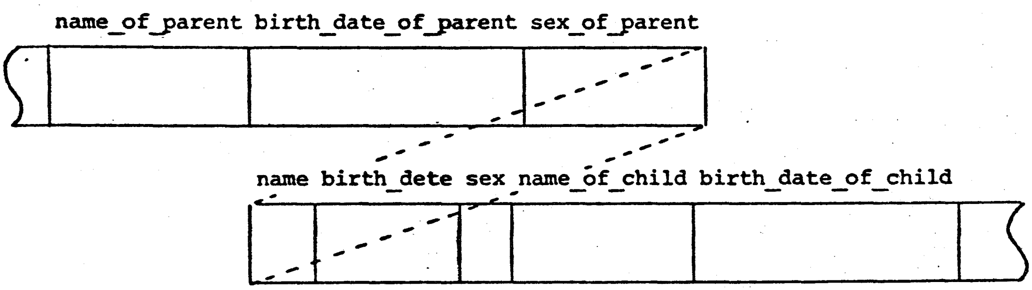
character	novel	author
	The adventure of Tom Sawyer	Mark Twain
Porfiry	Crime and Punishment	Feodor Dostoyevsky
Robert Jordan	For Whom the Bell Tolls	Ernest Hemingway
Pilar	For Whom the Bell Tolls	Ernest Hemingway
	Gone with the Wind	Magaret Mitchell
Ishmael	Moby-Dick	
Philip Carey	Of Human Bondage	

(b) an integrated view of two relations in (a)

Fig. 2. A synthetic interrelational relationship and an integrated view.

name	parent	birth date	sex
J. Smith	A. Smith	Dec. 11 1940	male
R. King	S. Brown	Mar. 20 1920	male
J. Smith	B. Wilson	Dec. 11 1940	male
P. Scott	L. Scott	May 9 1970	female
Y. Tanaka	K. Tanaka	Feb. 17 1950	male
A. Smith	T. Smith	Nov. 15 1915	male
B. Wilson	K. Wilson	Jun. 8 1918	female
H. King	R. King	Jul. 1 1950	male
:	:	:	:

(a) a relation with a recursive relationship.



(b) a view of (a) with infinitely many attributes.

Fig. 3. A recursive relationship and a view with infinitely many attributes.

(1) Find the name and the sex of a person whose birth date is Feb. 17 1950.

```
select   name, sex
where    birth date = 'Feb. 17 1950'.
```

(2) Find the name of the parent and the sex of a parson whose birth date is Feb. 17 1950.

```
select   parent, sex
where    birth date = 'Feb. 17 1950'.
```

(3) Find the name and sex of a person whose parent's birth date is Feb. 17 1950.

```
select   name, sex
where    parent  in
         select   name
         where    birth date = 'Feb. 17 1950'.
or
select   e1.name, e1.sex
where    e1.parent = e2.name
     and e2.birth date = 'Feb. 17 1950'.
```

(4) Find the name of the parent and the sex of a parson whose parent's birth date is Feb. 17 1950.

```
select   e2.name, e1.sex
where    e1.parent = e2. name
     and e2.birth date = 'Feb. 17 1950'.
```

(a) four queries written in English and a SEQUEL like language.

```
(1)   select   name, sex
      where    birth date = 'Feb. 17 1950'.
(2)   select   name of parent, sex
      where    birth date = 'Feb. 17 1950'.
(3)   select   name, sex
      where    birth date of parent = 'Feb. 17 1950'.
(4)   select   name of parent, sex
      where    birth date of parent = 'Feb. 17 1950'.
```

(b) queries based on the view in Fig.3 (b).

Fig. 4. Various queries of a database in Fig.3 (a) and those based on the view of this database shown in Fig.3 (b).

Fig.3 (a), the information about the antecedents of J. Smith forms a semantically meaningful subspace. It is very reasonable in some possible applications to restrict the access right of each user within information about his own antecedents. How can we formally specify this kind of subspaces? Relational model can not answer this question since every subspace describable by this model is a subpart of some single relation or a union of such subparts (Fig.5).

The third problem concerns the formal semantics of natural language vocabulary. If we can formally define the semantics of the phrase "of the parent" in Fig.3 (a), then we can also define the semantics of "of the father", "of the brother", "of the sister", etc. However, no single phrase of the latters defines the former. In this database, "of the parent" is an elementary synthetic relationship, while the others are derivable from this. It may be expected that we will be able to define formal semantics of various vocabularies from the semantics of elementary synthetic interrelational relationships.

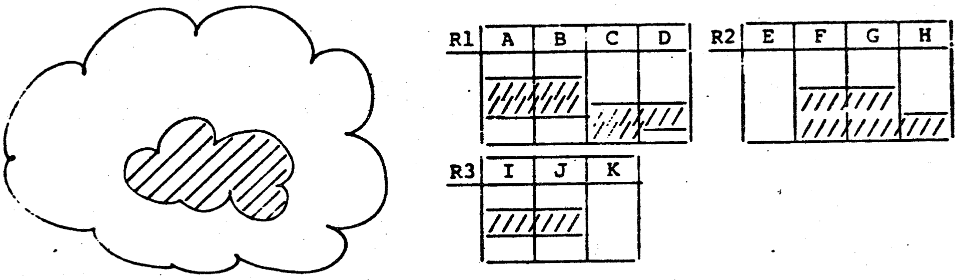
All these problems above concern synthetic structures among relations rather than analytic structures. They prove the importance of the formalization of synthetic interrelational relationships.

3.MORPHISM BETWEEN RELATIONS

3.1. Formal Interpretation of a Synthetic Interrelational Relationship.

Fig.6 shows an example of synthetic interrelational relationships. Suppose that R1 is a relation about the managemental information of an institute and R2 is a relation about bibliographic information for reference use in this institute. The relation R2 includes not only papers written by staffs in this institute but also those by authors outside of this institute. These two relations are related synthetically but not analytically. Integration of these two relations enables us to search papers written by a project in a focus. These are papers written by such authors who are staffs of this project. Such papers are "papers of the project". This adjective phrase "of the project" can be considered as a morphism, i.e., a relational morphism, that relates two relations R1 and R2, i.e., $\sigma:R2 \rightarrow R1$. This morphism induces a mapping that maps information about documents to information about documents of the project. The latter might be considered as a part of the information of the project.

As is shown in Fig.7, a morphism $\sigma:R2 \rightarrow R1$ relates two tuples p in R1 and d in R2 in such a way that /staff/-value of p is equal to /author/-value of d. It conceptually extend a tuple p to p' that is a concatenation of p and the image of d mapped by σ. An extended tuple p' represents information of the project in a focus. For an attribute A in R1, A-component of p' represents information about A of the project, while, for an attribute B in R2, B-component of p' represents information about B of a document of the project. Since obviously p' represents information of the project, we omit the last phrase "of the project" from the names of each component of p'.



(a) a subspace of an information space.

(b) subparts of relations.

Fig. 5. Difference between a subspace of an information space and a union of subrelations.

R1(/project/, /budget/, /staff/)

R2(/author/, /title/, /journal/)

Fig. 6. An example database with a synthetic interrelational relationship.

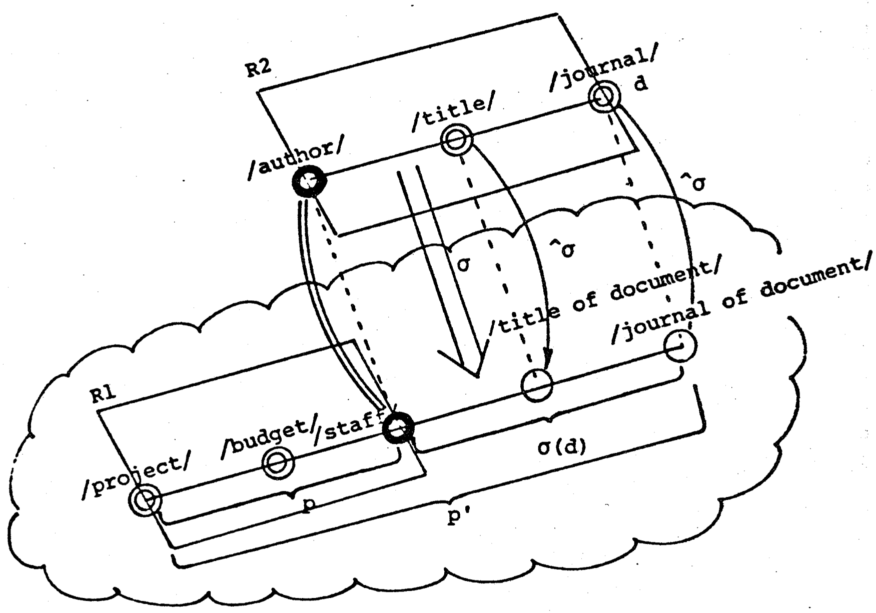


Fig. 7. Pictorial interpretation of an interrelational morphism.

Therefore, A and B-components of p' above can be denoted by 'A' and 'B of document' respectively. A set of such extended tuples as p' forms an information space represented by a cloud in Fig.7. Between each attribute B of R2 and its counterpart 'B of document' in this cloud, we can define a morphism $\hat{}\sigma$ such that $\hat{}\sigma$: B $\mapsto$ B of document. This morphism is called a labeling morphism since it labels B with "of document". Labeling morphisms should be one-to-one. We can define the morphism σ in the above example in such a way as

<u>morphism</u> σ: R2 $\rightarrow$ R1
<u>where</u> $\hat{}\sigma$(/author/)=/staff/.

For each morphism σ, we can define its reverse σ^-. For σ above, σ^- is equal to the following definition;

<u>morphism</u> σ^-: R1 $\rightarrow$ R2
<u>where</u> $\hat{}\sigma^-$(/staff/)=/author/.

3.2. Formal Theory of Morphisms and Semantic Attributes

In the sequel, we use the following notations;

(1) Pred(X), Pred_i(X) : a predicate about attributes in an attribute set X,
(2) [X]R : projection of a relation R to an attribute set X,
(3) [Pred(X)]R : restriction of a relation R with respect to the condition Pred(X)=true,
(4) RS : Cartesian product of two relations,
(5) $\cap$RS, $\cup$RS : intersection and union of two relations R and S with a same attribute set,
(6) $\bowtie$RS : natural join of R and S with respect to the common attributes,
(7) $\langle X\rangle_0$: a relation over an attribute set X,
(8) Ω(R) : the set of all the attributes of a relation R,
(9) $|S|$: the number of elements in a set S.

Let R denote all the first normal form relations of a database and M a set of morphisms in R. It is assumed that, for each morphism σ in M, a special morphism σ^- called a reverse of σ is also included in M, where $(\sigma^-)^-=\sigma$. We assume that each relation in R has different attributes disjoint from those of the other relations in R. This condition is always satisfied after proper renaming of attributes. The world W is a Cartesian product of all the relations in R, i.e.,

$$W = \Pi_{R \in \mathcal{R}} R. \tag{3.1}$$

We denote this by $\langle\Omega\rangle_0$, where

$$\Omega = \cup_{R \in \mathcal{R}} \Omega(R). \tag{3.2}$$

Let $l(\sigma)$ denote a label corresponding to a morphism σ such that the labeling morphism $\hat{}\sigma$ mapps any attribute A in Ω to $A\,l(\sigma)$. We define semantic attributes as follows;

(1) attributes in Ω are semantic attributes,
(2) if A is a semantic attribute and σ is in M, then $A\,l(\sigma)$ is a semantic attribute,
(3) only those obtained by finite applications of the above two rules are semantic attributes.

We denote the set of finite sequences of labels by $L^*(M)$ and the set of semantic attributes by Ω^*, i.e.,

$$\Omega^* = \{A\rho \mid A\in\Omega \text{ and } \rho\in L^*(M)\}. \tag{3.3}$$

By $X\rho$, we denote a set $\{A\rho \mid A\in X\}$ for any $X\in\Omega^*$ and $\rho\in L^*(M)$.

A morphism σ between relation R and S is defined by a statement:

$$\begin{array}{ll}\underline{\text{morphism}} & \sigma\colon R\rightarrow S\\ \underline{\text{where}} & \mathrm{Pred}(\hat{}\sigma(X),\ Y),\end{array} \tag{3.4}$$

where $X\subset\Omega(R)$, $Y\subset\Omega(S)$, and $\hat{}\sigma(X)=X\,l(\sigma)$. It is denoted by σRS that R and S are related by a morphism $\sigma\colon R\rightarrow S$. Relations R and S are called a domain relation and a codomain relation of σ.

The formal semantics of σ is defined by its natural extension $\hat{\sigma}\colon 2^{\Omega^*}\rightarrow 2^{\Omega^*}$ as

$$\sigma RS \equiv \hat{\sigma}\Omega(R)\Omega(S). \tag{3.5}$$

Let $\alpha(\rho)$ denote a unary operator that renames every semantic attribute A in the immediately following term to $A\rho$. The natural extension $\hat{\sigma}$ of σ is defined by a λ-expression:

$$\hat{\sigma} = \lambda xy.\ \langle (x\,l(\sigma))\cup y\rangle. \tag{3.6}$$

The relation $\langle x\rangle$ over an arbitrary subset x of Ω^* is recursively defined as follows;

(1) $\forall x\subset\Omega,\quad \langle x\rangle=\langle x\rangle_{\Omega}\ (=[x]\langle\Omega\rangle_{0}),$

(2) $\forall x,y\subset\Omega^*$ s.t. $y\cap(\Omega^* l(\sigma))=\phi,$

$$\begin{aligned}&\langle (x\,l(\sigma))\cup y\rangle\\ &=[(x\,l(\sigma))\cup y][\mathrm{Pred}(X\,l(\sigma),\ Y)](\alpha(l(\sigma))\langle x\cup X\rangle\langle y\cup Y\rangle),\end{aligned} \tag{3.7}$$

(3) $\forall x\subset\Omega^*,\ \langle x\,l(\sigma)\rangle=\alpha(l(\sigma))\langle x\rangle.$

Let σ and τ be two morphisms in M defined as

$$\begin{array}{ll}\underline{\text{morphism}} & \sigma : R1\rightarrow S1\\ \underline{\text{where}} & \mathrm{Pred}_1(\hat{}\sigma(X_1),\ Y_1)\\ & (\ X_1\subset\Omega(R1),\ Y_1\subset\Omega(S1)\ )\end{array}$$

and

$\underline{\text{morphism}}$ $\tau : R2 \rightarrow S2$
$\underline{\text{where}}$ $Pred_2(\hat{}\sigma(X_2), Y_2)$
$(X_2 \subset \Omega(R2), Y_2 \subset \Omega(S2))$.

The composite morphism $\tau\sigma$ is defined as

$$(\tau\sigma)(R1)(S2) \equiv (\widehat{\tau\sigma})\Omega(R1)\Omega(S2),$$

where $\mathcal{l}(\tau\sigma)$ is defined to be $\mathcal{l}(\tau)\,\mathcal{l}(\sigma)$. Independently from the above definition, we define the composition of the natural extensions $\hat{\sigma}$ and $\hat{\tau}$ as

$$\hat{\tau}\cdot\hat{\sigma} = \lambda xy.\ [(x\,\mathcal{l}(\tau)\,\mathcal{l}(\sigma))^{\vee} y]\bowtie\alpha(\mathcal{l}(\sigma))(\hat{\tau}x\Omega)(\hat{\sigma}\Omega y). \quad (3.8)$$

Then the following theorem holds.

<u>Theorem 3.1</u>
For any σ, τ in M, it holds that

$$\widehat{\tau\sigma} = \hat{\tau}\cdot\hat{\sigma}. \quad (3.9)$$

<u>proof</u>
Since it holds that

$$\hat{\tau}x\Omega = <(x\,\mathcal{l}(\tau))^{\vee}\,\Omega>$$

and

$$\hat{\sigma}\Omega y = <(\Omega\,\mathcal{l}(\sigma))^{\vee}\,y>,$$

the following equalities hold;

$$\begin{aligned}
&\bowtie\alpha(\mathcal{l}(\sigma))(\hat{\tau}x\Omega)(\hat{\sigma}\Omega y)\\
&=\bowtie<(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}(\Omega\mathcal{l}(\sigma))><(\Omega\mathcal{l}(\sigma))^{\vee}y>\\
&=<(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}(\Omega\mathcal{l}(\sigma))^{\vee}y>.
\end{aligned}$$

Hence, the theorem is proved as follows;

$$\begin{aligned}
&(\hat{\tau}\cdot\hat{\sigma})xy\\
&=[(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}y]\bowtie\alpha(\mathcal{l}(\sigma))(\hat{\tau}x\Omega)(\hat{\sigma}\Omega y)\\
&=[(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}y]<(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}(\Omega\mathcal{l}(\sigma))^{\vee}y>\\
&=<(x\mathcal{l}(\tau)\mathcal{l}(\sigma))^{\vee}y>\\
&=<(x\mathcal{l}(\tau\sigma))^{\vee}y>\\
&=\widehat{\tau\sigma}xy.
\end{aligned}$$

Now we extend the definition of a labeling morphism $\hat{}\sigma$ as $\hat{}\sigma : 2^{\Omega^*} \rightarrow 2^{\Omega^*}$.
From the fact that $\mathcal{l}(\tau\sigma)=\mathcal{l}(\tau)\mathcal{l}(\sigma)$, it should be defined as

$$\hat{}\sigma(x\rho) = x\mathcal{l}(\sigma)\rho \text{ for any subset } x \text{ of } \Omega^* \text{ and any } \rho \text{ in } L^*(M). \quad (3.10)$$

The identity morphism $\hat{1}$ in 2^{Ω^*} is defined as

$$\hat{1} = \lambda xy.\ <x^{\vee}y>. \quad (3.11)$$

For a morphism σ defined by (3.4), we define its reverse σ^{-} as follows;

<u>morphism</u> $\sigma^{-} : S \rightarrow R_{-}$
<u>where</u> $\mathrm{Pred}(X, {}^{\wedge}\sigma^{-}(Y))$. (3.12)

It should be noticed that $\widehat{\sigma^{-}} \cdot \hat{\partial}=1$ does not always hold. In fact, it holds if and only if

(1) $\forall x \in \langle X \rangle,\ \exists y \in \langle Y \rangle \quad \mathrm{Pred}(x, y)=\mathrm{true}$

and

(2) $\forall x, x' \in \langle X \rangle,\ \forall y \in \langle Y \rangle$
$(\ \mathrm{Pred}(x, y) \wedge \mathrm{Pred}(x', y)\ ) \supset (\ x=x'\ )$.

Let $(l_1/l_2, l_3/l_4, \ldots l_{2n-1}/l_{2n})$ be a unary operator that renames every semantic attribute $A\rho l_{2i}$ in the immediately following term to $A\rho l_{2i-1}$. Let σ and τ be same as before. The conjunction $\Lambda\sigma\tau$ of σ and τ is defined as

$$\widehat{\Lambda\sigma\tau} = \Lambda\hat{\partial}\hat{\tau}$$
$$= \lambda xy.\ [(x l(\Lambda\sigma\tau)) \cup y]$$
$$\cap\ (l(\Lambda\sigma\tau)/l(\sigma))(\hat{\partial}xy)(l(\Lambda\sigma\tau)/l(\tau))(\hat{\tau}xy), \qquad (3.13)$$

while the disjunction $V\sigma\tau$ is defined as

$$\widehat{V\sigma\tau} = V\hat{\partial}\hat{\tau}$$
$$= \lambda xy.\ [(x l(V\sigma\tau)) \cup y]$$
$$\cup\ (l(V\sigma\tau)/l(\sigma))(\hat{\partial}xy)(l(V\sigma\tau)/l(\tau))(\hat{\tau}xy). \qquad (3.14)$$

Here we also extend the definition of a restriction operator [Pred(X)] as

$$[\mathrm{Pred}(X)] = \lambda x.\ [x][\mathrm{Pred}(X)]\langle x \cup X \rangle. \qquad (3.15)$$

<u>Example 3.1</u>

For a database with a single relation

R(/name/, /parent/, /birth date/, /sex/),

we can define the following morphisms, where $w(\sigma)$ denotes the English word such that "of $w(\sigma)$" corresponds to $l(\sigma)$.

(1) $w(\sigma 1)$ = 'parent', $w(\sigma 1^{-})$ = 'child'

<u>morphism</u> $\sigma 1 : R \rightarrow R$
<u>where</u> ${}^{\wedge}\sigma 1$(/name/)=/parent/.

(2) $w(\sigma 2)$ = 'father'

<u>morphism</u> $\sigma 2 : R \rightarrow R$
<u>where</u> (${}^{\wedge}\sigma 2$(/name/)=/parent/) $\wedge$ (${}^{\wedge}\sigma 2$(/sex/)='male').

(3) $w(\sigma 3)$ = 'mother'

<u>morphism</u> $\sigma 3$; $R \to R$
<u>where</u> $(\hat{}\sigma 3(/name/)=/parent/) \wedge (\hat{}\sigma 3(/sex/)='female')$.

The morphisms $\sigma 2$ and $\sigma 3$ can be defined by $\sigma 1$, i.e.,

$\hat{\partial}2$ = [/sex/$l(\sigma 1)$='male']$\hat{\partial}1$,

$\hat{\partial}3$ = [/sex/$l(\sigma 1)$='female']$\hat{\partial}1$.

On the other hand, it holds that

$\hat{\partial}1 = \vee\hat{\partial}2\hat{\partial}3$.

With $\sigma 1$ and $\sigma 1^{-}$, we can define various English words as composition of these morphisms.

(4) $w(\sigma 4)$ = 'son'

$\hat{\partial}4$ = [/sex/$l(\sigma 1^{-})$='male']$\hat{\partial}1^{-}$.

(5) $w(\sigma 5)$ = 'daughter'

$\hat{\partial}5$ = [/sex/$l(\sigma 1^{-})$='female']$\hat{\partial}1^{-}$.

(6) $w(\sigma 6)$ = 'brother'

$\hat{\partial}6$ = [/sex/$l(\sigma 1^{-})l(\sigma 1)$='male']$(\hat{\partial}1^{-}\hat{\partial}1)$.

If a boy is not considered as a brother of himself, then $\sigma 6$ is expressed as follows, where diff(R)(S) denotes set difference of two relations with a same attribute set.

$\hat{\partial}6$ = λxy.diff [/sex/$l(\sigma 1^{-})l(\sigma 1)$='male']$(\hat{\partial}1^{-}\hat{\partial}1)$xy $\hat{1}$xy

(7) $w(\sigma 7)$ = 'grandfather'

$\hat{\partial}7$ = [/sex/$l(\sigma 1)l(\sigma 1)$='male']$\hat{\partial}1\hat{\partial}1$.

These examples give answers to the third questions in section 2.2.

<u>Example 3.2</u>

Suppose we have the following two relations:

R1(/project/, /staff/, /budget/)

R2(/title/, /author/, /journal/)

We can define the following morphism:

$w(\sigma)$ = 'document', $w(\sigma^{-})$ = 'project'

<u>morphism</u> σ : $R2 \to R1$
<u>where</u> $\hat{}\sigma$(/author/)=/staff/.

In this database, the composite morphism $\sigma\sigma$ or $\sigma^{-}\sigma^{-}$ is nonsense. However, we do not prohibit the use of these composite morphisms since it is not harmful to formally define these. For example, $\hat{\partial}\hat{\partial}${/title/}{/project/} is defined as follows from the definition of morphisms and their interpretation.

∂∂{/title/}{/project/}

=</title/ℓ(σ)ℓ(σ), /project/>

=[/staff/1=/author/1]</staff/1, /project/></author/1, /title/ℓ(σ)>

=[/staff/1=/author/1]([/staff/1, /project/]R1)

([/staff/2=/author/2]</author/1, /staff/2></author/2, /title/>)

=[/staff/1=/author/1]([/staff/1, /project/]R1)

([/staff/2=/author/2]([/author/1, /staff/2]R1R2)

([/author/2, /title/]R2)).

Since a cartesian product R1R2 appears at the end of the second line of the last transformation and no interrelational restriction is specified between R1 and R2, the /title/ and the /project/ in ∂∂{/title/}{/project/} do not have any significant relationship between themselves.

The composition $\tau\sigma$ is meaningless iff the codomain relation of σ is equal to the domain relation of τ.

Example 3.3

In the following database, there are subordinate relationships between attributes, i.e., drivers and secretaries are also employees.

R1(/employee/, /salary/, /address/)

R2(/driver/, /license no./)

R3(/secretary/, /typing speed/)

/employee/

/driver/ /secretary/

These subordinate relationships are also represented by morphisms below.

σ1 morphism σ1 : R2 ⟶ R1
where ^σ1(/driver/)=/employee/.

σ2 morphism σ2 : R3 ⟶ R1
where ^σ2(/secretary/)=/employee/.

We will return to these subordinate relationships afterwards in section 7.

4. INFORMATION SPACE MODEL

4.1. Elementary Morphism

We say a morphism σ is elementary if it is defined in the following form;

morphism σ : R ⟶ S
where ^σ(A)=B, (4.1)

where A and B are elements of $\Omega(R)$ and $\Omega(S)$ respectively. In most of the applications, the most general form of morphism definitions may be as follows;

$$\underline{\text{morphism}}\quad \sigma : R \rightarrow S$$
$$\underline{\text{where}}\ (\hat{\ }\sigma(X_1)=Y_1) \wedge \text{Pred}_1(\hat{\ }\sigma(X_2)) \wedge \text{Pred}_2(Y_2), \tag{4.2}$$
$$(\ X_i \subset \Omega(R),\ Y_i \subset \Omega(S),\ |X_1|=|Y_1|\ ).$$

For such morphisms, the following theorem holds.

Theorem 4.1

Any morphism with the form

$$\underline{\text{morphism}}\quad \sigma : R \rightarrow S$$
$$\underline{\text{where}}\quad \bigwedge_{i=1}^{k} (\hat{\ }\sigma(A_i)=B_i) \wedge \text{Pred}_1(\hat{\ }\sigma(X)) \wedge \text{Pred}_2(Y) \tag{4.3}$$
$$(\ A_i \in \Omega(R),\quad B_i \in \Omega(S),\quad X \subset \Omega(R),\quad Y \subset \Omega(S)\ )$$

can be defined using elementary morphisms.

proof

Let σ_i denote an elementary morphism defined as

$$\underline{\text{morphism}}\quad \sigma_i : R \rightarrow S$$
$$\underline{\text{where}}\quad \hat{\ }\sigma_i(A_i)=B_i. \tag{4.4}$$

Then it holds that

$$\hat{\sigma} = [\text{Pred}_1(\hat{\ }\tau(X))][\text{Pred}_2(Y)]\hat{\tau}, \tag{4.5}$$

where

$$\hat{\tau} = \bigwedge_{i=1}^{k} \hat{\sigma}_i. \tag{4.6}$$

This theorem indicates that only a set of elementary morphisms is sufficient to describe synthetic interrelational relationships in an object database. A set of elementary morphisms from which any morphism in M can be derived is denoted by M_0.

For a given set R of relations, a set M of morphisms is said to be sufficient if any synthetic interrelational relationships in R can be represented by elements of this set. A pair (R, M) is called an information space schema if M is sufficient with respect to R. In most of the applications, a schema (R, M) has an equivalent schema (R, M_0), where M_0 is a set of elementary morphisms. The schema (R, M_0) is called a normal form schema of (R, M). It should be noticed that the number of elementary morphisms is always finite. Therefore, we can always define information space schema with finite description.

4.2. World and View Point

For each ρ in $L^*(M_0)$, the set $\Omega\rho$ forms a world of information labeled with ρ.

We denote this world by $W\rho$. Let f_{σ} be defined as

$$f_{\sigma} = \lambda x.\ \hat{\sigma} x. \qquad (4.7)$$

The morphism f_{σ} is interpreted as a view point shifter that moves the view point from the world $W\ell(\sigma)\rho$ to $W\rho$. The world W is especially called a base world.

Example 4.1

$I = (R, M_0)$

$R = \{R1, R2\}$

R1(/project/,/budget/,/manager/,/employee/,/salary/,/department/, /location/,/subproject/)

R2(/report no./,/title/,/author/,/journal/,/key word/)

$M_0 = \{\sigma 1, \sigma 1^-, \sigma 2, \sigma 2^-, \sigma 3, \sigma 3^-\}$

$\sigma 1$ $w(\sigma 1)$ = 'project', $w(\sigma 1^-)$ = 'document'

morphism $\sigma 1 : R1 \rightarrow R2$
where $\hat{\ }\sigma 1$(/employee/)=/author/

$\sigma 2$ $w(\sigma 2)$ = 'manager', $w(\sigma 2^-)$ = 'subordinate'

morphism $\sigma 2 : R1 \rightarrow R1$
where $\hat{\ }\sigma 2$(/employee/)=/manager/

$\sigma 3$ $w(\sigma 3)$ = 'subproject', $w(\sigma 3^-)$ = 'superproject'

morphism $\sigma 3 : R1 \rightarrow R1$
where $\hat{\ }\sigma 3$(/project/)=/subproject/

The diagramatic representation of this schema is shown in Fig.8. In Fig.9, we show the pictorial representation of the relationships among worlds. An eye in Fig.9 indicates the view point.

4.3. Formal Description of an Information Subspace

By semantic subspace, we mean a relation over a subset X of semantic attributes that satisfies the condition Pred(Y), where Y is also a subset of semantic attributes. Let this subspace be named W. Then W is defined as

$$W = [X][Pred(Y)]\langle X \cup Y \rangle. \qquad (4.8)$$

We formally describe W as

S-subspace W
over X
where Pred(Y). (4.9)

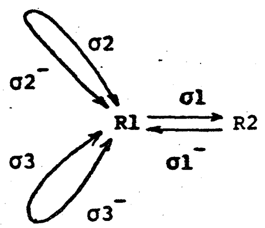
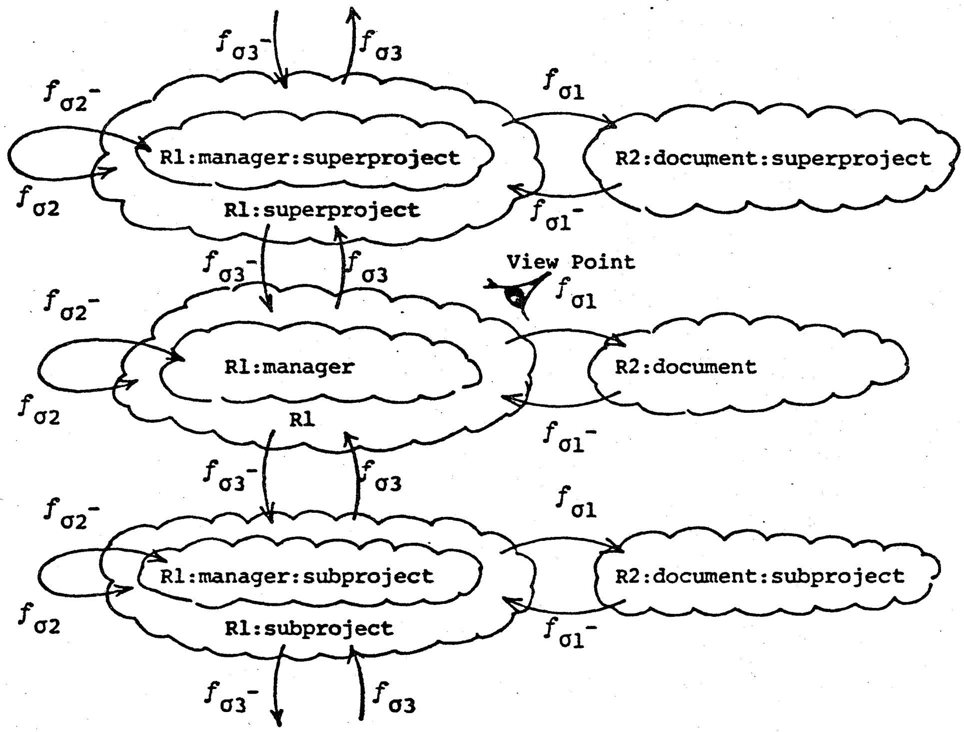


Fig. 8. Diagramatic representation of the schema in example 4.1.



It is assumed that R1 is in a focus.
Meanigless composition of morphisms is neglected.

Fig. 9. Pictorial representation of the relatioships among worlds.

Let W_1 and W_2 be a semantic subspace. Then the intersection of these relations is equal to a relation W defined as

$$W = W_1 \cap W_2$$
$$= \cap([X_1][Pred_1(Y_1)]<X_1 \cup Y_1>)([X_2][Pred_2(Y_2)]<X_2 \cup Y_2>)$$
$$= [X_1 \cap X_2][Pred_1(Y_1) \wedge Pred_2(Y_2)]<X_1 \cup Y_1 \cup X_2 \cup Y_2>.$$

Therefore W is also a semantic subspace described as

<u>S-space</u> W
<u>over</u> $X_1 \cap X_2$
<u>where</u> $Pred_1(Y_1) \wedge Pred_2(Y_2)$. (4.10)

However, the union of two semantic subspaces can not be described as (4.9). Hence, $W_1 \cup W_2$ is not a semantic subspace unless one of the followings holds;

(1) $W_1 \supset W_2$,

(2) $W_2 \supset W_1$,

(3) $X_1 = X_2$,

(4) $Pred_1(Y_1) = Pred_2(Y_2)$.

We define an information subspace as follows;

(1) a semantic subspace is an information subspace,

(2) if W_1 and W_2 are information subspaces then $W_1 \cup W_2$ is an information subspace.

(3) if W_1 and W_2 are information subspaces then $W_1 \cap W_2$ is an information subspace.

(4) only those obtained by finite applications of the above rules are information subspaces.

This definition of an information subspace is a very natural formal definition of a meaningful subspace of information mentioned in section 2.2. The above observation indicates that any information subspace can be defined as a union of finite number of semantic subspaces in such a way as follows;

<u>I-subspace</u> W
<u>where</u> $W = \cup_{i \in I} W_i$, (4.11)

for every $i \in I$,

<u>S-subspace</u> W_i
<u>over</u> X_i
<u>where</u> $Pred_i(Y_i)$. (4.12)

For example, in a database in example 3.1, the information about the antecedents of J. Smith and that about the descendants of R. King forms the information subspace W described below;

<u>I-subspace</u> W
<u>where</u> $W = W_1 \cup W_2$,
<u>S-subspace</u> W_1
<u>over</u> /name/$(l(\sigma_1))^*$, /birth date/$(l(\sigma_1))^*$, /sex/$(l(\sigma_1))^*$
<u>where</u> /name/ = 'J. Smith',
<u>S-subspace</u> W_2
<u>over</u> /name/$(l(\sigma_1^-))^*$, /birth date/$(l(\sigma_1^-))^*$, /sex/$(l(\sigma_1^-))^*$
<u>where</u> /name/ = 'R. King',

where Al^* denotes a set of all the semantic attributes $A\rho$ such that ρ is a list of finitely iterated l.

This facility to define information subspaces enhances the capability of the database management system in access control by query modifications proposed in [ASTR76][CHAM76]. This problem as well as other applications of this facility will be reported elsewhere.

5. RECURSIVE MORPHISM AND DIRECT SUM DECOMPOSITION

In this section, we further investigate semantic structures induced by recursive morphisms. We restrict our discussion to such a case with only one first normal form relation. Since this section deals only with the decomposition of a relation with a recursive morphism and a recursive morphism is defined within a single relation, the following result is also applicable to the cases with more than one relations.

Let R be an object first normal form relation. A recursive attribute is defined if there exists two attributes A and B in the attribute set $\Omega(R)$, such that

(1) the domains of these two attributes Dom(A) and Dom(B) intersect with each other,
(2) we can assume for any $x \in$ Dom(A) x is also a member of Dom(B) without any contradiction and vice versa,
(3) either of the following two MVDs holds;

$$A \twoheadrightarrow \Omega_1 \mid \Omega_2 B,$$
$$B \twoheadrightarrow \Omega_1 \mid \Omega_2 A,$$

where $\{A\}$, $\{B\}$, Ω_1, Ω_2 are partition of $\Omega(R)$.

A pair of these attributes A and B is called a recursive attribute pair. Suppose that A and B is a recursive attribute pair satisfying $A \twoheadrightarrow \Omega_1 \mid \Omega_2$ B, where $\{A\}$, $\{B\}$, Ω_1, Ω_2 form a partition of $\Omega(R)$. Then B is called a recursive attribute, and A a superordinate attribute. We introduce a new attribute B^a corresponding to the antonym of B. The superordinate attribute of B is denoted by B^s.

For example, /name/ and /parent/ form a recursive attribute pair in example 3.1. Since it holds that

$$/name/ \twoheadrightarrow /birth\ date//sex/ \mid /parent/$$

but that

$$/parent/ \twoheadrightarrow /name//birth\ date//sex/,$$

/parent/ is a recursive attribute. We can introduce /child/ as an antonym of /parent/.

Suppose that there exists no such subset Ω' of Ω that satisfies an MVD

$$\phi \twoheadrightarrow \Omega' \ \text{ in } \Omega,$$

where ϕ denotes an empty set. If there exists one, then we can apply the following result to Ω' and $\Omega(R)-\Omega'$ independently because R is a Cartesian product of these in such a case.

Suppose that there exists h recursive attributes B_i $(1 \leq i \leq h)$. Let Ω° denote

$$\Omega^\circ = \Omega \cup (\cup_{1\leq i\leq h} \{B_i, B_i^a, B_i^s\}).$$

We call the following condition an S-condition;

$$\{B_i^s \mid 1\leq i\leq h\} \twoheadrightarrow \{B_i\} \ \text{ in } \ \{B_i^s, B_i \mid 1\leq i\leq h\}.$$

Suppose that a set of h recursive attributes $\{B_i \mid 1\leq i\leq h\}$ in Ω satisfies S-condition. Let Ω^+ denote $\Omega^\circ-(\cup_{1\leq i\leq h} \{B_i^a\})$, and Ω^+_i be a minimal subset Ω' including B_i such that

$$\{B_i^s\} \twoheadrightarrow \Omega' \ \text{ in } \Omega^+.$$

Let Ω_i be defined as

$$\Omega_i = \Omega^+_i \cup \{B_i^a\} \quad (1\leq i\leq h),$$

$$\Omega_0 = \Omega^\circ-(\cup_{1\leq i\leq h} \Omega_i).$$

Then Ω° is represented as a direct sum of Ω_i $(0\leq i\leq h)$.

<u>Theorem 5.1</u>

$$\Omega^\circ = \Omega_0 \oplus \Omega_1 \oplus \ldots \oplus \Omega_h. \tag{5.1}$$

This means that $\{\Omega_i \mid 0\leq i\leq h\}$ is a partition of Ω°. Fig.10 shows an example relation with 2 recursive attributes and its direct sum decomposition.

We can define a recursive morphism σ_i for each recursive attribute pair (A_i, B_i)

attribute set :

Ω = {/person/,/project/,/section/,/department/,/company/,/subsidiary/,
/location/,/subproject/,/subproject-name/}

We assume that a project is called by different names under different superprojects.

recursive attributes :

A_1 = /subsidiary/

A_2 = /subproject/

$h = 2$

antonym :

A_1^a = /parent company/

A_2^a = /superproject/

superordinate attributes :

A_1^s = /company/

A_2^s = /project/

$\Omega° = \Omega \cup$ {/parent company/, /superproject/}

$\Omega^+ = \Omega$

S-condition :

{/company/,/project/} $\twoheadrightarrow$ {/subsidiary/} | {/subproject/}

in {/company/,/project/,/subsidiary/,/subproject/}

direct sum decomposition :

Ω_1 : {/company/} $\twoheadrightarrow$ {/subsidiary/} in Ω^+

Ω_1 = {/subsidiary/, /parent company/}

Ω_2 : {/project/} $\twoheadrightarrow${/subproject/,/subproject-name/} in Ω^+

Ω_2 = {/subproject/,/superproject/,/subproject-name/}

Ω_0 : $\Omega_0 = \Omega° - \Omega_1 - \Omega_2$

= {/person/,/project/,/section/,/department/,/company/,/location/}

Fig. 10. Direct sum decomposition of a relation with two recursive attributes.

as follows, where B_i is assumed to be a recursive attribute.

$$\underline{\text{morphism}} \quad \sigma_i : R \rightarrow R$$
$$\underline{\text{where}} \quad A_i l(\sigma_i) = B_i. \tag{5.2}$$

It is recommended by various reports that the information about a recursive pair (A_i, B_i) should be separated from the rest of R. This is done by decomposing $\langle\Omega^\circ\rangle$ into $\{\langle\Omega_i\rangle \mid 0 \le i \le h\}$. If some Ω_i has a set of recursive attribute pairs satisfying S-condition then Ω_i is further decomposed by the direct sum decomposition method mentioned above. The original relation $\langle\Omega\rangle$ is related to $\{\langle\Omega_i\rangle\}$ by the following relation;

$$\langle\Omega\rangle = [\Omega][\bigwedge_{0 \le i \le h} (B_i{}^s = B_i{}^a)]\Pi\langle\Omega_i\rangle. \tag{5.3}$$

6. DESIGN OF AN INFORMATION SPACE MODEL

Suppose that set R of first normal form relations are given, and that, for any two different relations R and S in R, the attribute sets of these are mutually disjoint. This condition is always satisfiable by proper renaming of attributes.

The procedure for the design of an information space schema for R is summarized below.

(1) $W = \Pi_{R \in R} R$ (the base world).

(2) For each R in R, find out a set of recursive attributes $\{B_i \mid 1 \le i \le h\}$ satisfying S-condition and decompose R by the direct sum decomposition method. For each component of the decomposition of R, apply this step recursively until all components can not be further decomposed. Define a set P of recursive morphisms each of which corresponds to some recursive attribute found by this step.

(3) Find out other elementary morphisms in R. Let E denote a set of them. Let M_0 be the union of P and E. A pair (R, M_0) is the designed information space schema.

After the third step, we can apply our 4NF D-tree schema theory [TANA79] to decompose each first normal form relation in R into the fourth normal form relations. D-tree schema theory gives us the clear description about analytic interrelational relationships among the fourth normal form relations obtained by the decomposition.

7. QUERY LANGUAGE AND VOCABULARIES

Queries using semantic attributes can be described in the following form;

<u>select</u> X
<u>where</u> Pred(Y),

where X and Y denotes subsets of Ω^*. The execution of this query corresponds to the evaluation of

$$[X][\mathrm{Pred}(Y)]\langle X \vee Y\rangle .$$

The relation $\langle X \vee Y\rangle$ can be evaluated following the definition in section 3.2.

However, queries using semantic attributes are not sufficient to make them easy to understand.

Consider the database in Fig.2 (a). This has two relations below;

R1(/novel 1/, /author/)

R2(/character/, /novel 2/).

In this database, there are two morphisms σ and σ^{-} defined as

<u>morphism</u> $\sigma : R1 \rightarrow R2$
<u>where</u> $\hat{}\sigma$(/novel 1/) = /novel 2/.

In this case, it is very difficult to find out a proper adjective phrase for $\ell(\sigma)$. To solve this problem, we define vocabularies used in queries of this database with attribute names and morphisms. This is done as follows;

author ::= /author/$\ell(\sigma)$,
novel ::= /novel 1/$\ell(\sigma)$,
character ::= /character/.

Queries are written with these vocabularies. They are translated into the right hand sides of definitions by a query translator.

However, the definition of a word 'novel' as above may lead to wrong evaluation. Consider a query:

<u>select</u> novel, character.

This is evaluated as

$$
\begin{aligned}
&\langle \text{novel, character}\rangle \\
&=\langle \text{/novel 1/}\ell(\sigma)\text{, /character/}\rangle \\
&=[\text{/novel 1/}\ell(\sigma)\text{, /character/}][\text{/novel 1/}\ell(\sigma)\text{=/novel 2/}] \\
&\quad \alpha(\ell(\sigma))\langle\text{/novel 1/}\rangle\langle\text{/novel 2/, /character/}\rangle \\
&=[\text{/novel 1/}\ell(\sigma)\text{, /character/}][\text{/novel 1/}\ell(\sigma)\text{=/novel 2/}] \\
&\quad \alpha(\ell(\sigma))[\text{/novel 1/}]\text{R1 R2.}
\end{aligned}
$$

This is not equal to the desired result R2. Same is true with respect to the definition:

novel ::= /novel 2/.

This problem occurs if two attributes A in R and B in S are related and either of <A> or <B> is a subset of the other. This is solved by considering a new relation R0 that is a unary relation ⋃<A><B>. Then the new information space schema of this database becomes as follows;

R = {R0, R1, R2}

R0(/novel/) = ⋃[/novel 1/]R1 [/novel 2/]R2
R1(/novel 1/, /author/)
R2(/character/, /novel 2/),

M_0 = {σ1, σ2}

<u>morphism</u> σ1 : R1 → R0
<u>where</u> ^σ1(/novel 1/) = /novel/

<u>morphism</u> σ2 : R2 → R0
<u>where</u> ^σ2(/novel 2/) = /novel/.

The vocabularies are defined as

author ::= /author/ℓ(σ1),

novel ::= /novel/,

character ::= /character/ℓ(σ2).

The query is evaluated as

<novel, character>
=</novel/, /character/ℓ(σ2)
=[/novel/, /character/ℓ(σ2)][/novel 2/ℓ(σ2)=/novel/]
α(ℓ(σ2))</novel 2/, /character/>R0
=α(ℓ(σ2))</novel 2/, /character/>
=α(ℓ(σ2))R2.

<u>Example 7.1</u>

Consider the database with subordinate relationships between attributes shown in example 3.3. Since it holds that

</employee/> ⊃ </driver/>,

and

</employee/> ⊃ </secretary/>,

we can define the vocabularies of this database as follows;

employee ::= /employee/,

salary ::= /salary/,

address ::= /address/,

driver ::= /driver/ℓ(σ1),

license no. ::= /licence no./ℓ(σ1),

secretary ::= /secretary/$l(\sigma 2)$,

typing speed ::= /typing speed/$l(\sigma 2)$.

Example 7.2

The final example is a case with recursive morphisms shown in example 3.1. The vocabularies of this database are as follows;

name ::= /name/
parent ::= /parent/
birth date ::= /birth date/
sex ::= /sex/
father ::= /name/$l(\sigma 2)$
mother ::= /name/$l(\sigma 3)$
child ::= /name/$l(\sigma 1^{-})$
son ::= /name/$l(\sigma 4)$
daughter ::= /name/$l(\sigma 5)$
grandparent ::= /parent/$l(\sigma 1)$
grandfather ::= /name/$l(\sigma 2)l(\sigma 1)$
grandmother ::= /name/$l(\sigma 3)l(\sigma 1)$

⋮

descendant ::= /name/$(l(\sigma 1^{-}))^{*}$
antecedent ::= /name/$(l(\sigma 1))^{*}$

of parent ::= $l(\sigma 1)$
of father ::= $l(\sigma 2)$
of mother ::= $l(\sigma 3)$
of child ::= $l(\sigma 1^{-})$

⋮

As shown above, the vocabularies of a database consist of the noun definition and adjective definition. The detail formalization of vocabularies is reported elsewhere.

8. CONCLUDING REMARKS

While the relational model has been an infological framework of database theories, the information space model in this paper has been proposed as an infosemantic framework of database theories. Various semantic problems need theoretical basis for semantics, especially interrelational semantics. The idea of this model is very simple, i.e., a pair of R and M_0. The model is sufficient to solve various semantic problems shown in section 2.2.

The information space model should not be confused with the studies of functional programming in data bases [BUNE79][SHIP79]. Their main concern is the query program manipulating information. The information space model concerns the description of infosemantic structures of a schema as well as the improvement of query languages. While our model can cope with query programming problems as shown in section 7, recent studies on functional query language can not cope with the general description of information structures. Especially, they can not describe meaningful subspaces of information.

The use of a dictionary that defines nouns and adjectives from attribute names and morphisms may be a new approach to database semantics. This approach is enabled by the finiteness of the definition of an information space schema. We call this approach a denotational semantic approach to database semantics.

Reference

[ASTR76] M.M. Astrahan et al, "System R: A Relational Approach to Data Base Management," ACM Trans. on Database Systems, Vol.1, No. 2 (June 1976).

[BUNE79] P. Buneman, R.E. Frankel, "FQL--A Functional Query Language," Proc. of ACM-SIGMOD 1979, Boston, May 1979, pp.52-59.

[CHAM76] D.D. Chamberlin et al, "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control," IBM J. of R & D, Vol. 20, No. 6 (Nov. 1976).

[SHIP79] D. Shipman, "The Functional Data Model and the Data Language DAPLEX," Proc. of ACM-SIGMOD 1979, Boston, May 1979.

[TANA77] Y. Tanaka, T. Tsuda, "Decomposition and Composition of a Relational Database," Proc. of 3rd VLDB Conf., Tokyo, Oct. 1977, pp.454-461.

[TANA79] Y. Tanaka, "Logical Design of a Relational Schema and Integrity of a Data Base," Proc. of IFIP TC-2 Working Conf. on Data Base Architecture, also in *Data Base Architecture* (G. Bracchi, G.M. Nijssen ed.), North-Holland, 1979, pp.297-316.