közlemények

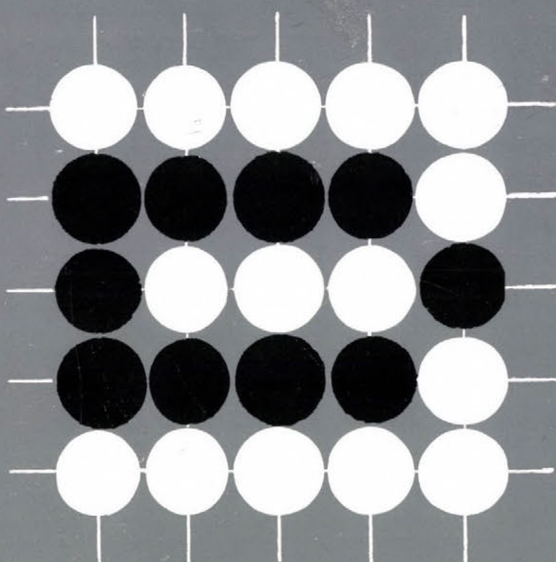
55807

30/1984

JATE KÖZPONTI KÖNYVTÁRA
SZEGED

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest

1985 SEP 2



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

K Ö Z L E M É N Y E K

30/1984

Szerkesztőbizottság:

GERTLER JÁNOS (felelős szerkesztő)
DEMETROVICS JÁNOS (titkár)
BACH IVÁN, UHRIN BÉLA, KERESZTÉLY SÁNDOR,
KNUTH ELŐD, KRÁMLI ANDRÁS, PRÉKOPA ANDRÁS,
ARATÓ MÁTYÁS

Felelős kiadó:

DR. VÁMOS TIBOR

ISBN 963 311 176 5
ISSN 0133-7459

Coopinvest 84.151

TARTALOMJEGYZÉK

CONTENTS

СОДЕРЖАНИЕ

Közlemények 30/1984 7-36

TRANSLATIONS OF RELATIONAL SCHEMAS

HO THUAN and *LE VAN BAO*

Computer and Automation Institute,
Hungarian Academy of Sciences

Institute of Computer Sciences and Cybernetics
Hanoi, Vietnam

INTRODUCTION

In this paper we shall be concerned with a class of translations of relational schemas.

Starting from a given relational schema, translations make it possible to obtain simpler schemas, i.e. those with a less number of attributes and with shorter functional dependencies so that the key-finding problem becomes less cumbersome, etc.

On the other hand, from the set of keys of the run relational schema obtained in this way the corresponding keys of the original schema can be found by a single "translation".

In §1 we introduce the notion of z-translation of a relational schema, give a classification of the relational schemas and inverstigate the characteristic properties of some classes of z-transformations.

In §2 we study some properties of the so called nontranslatable relational schemas.

The notation used here is the same as in [1]; $\subset$ means strict inclusion.

§1.

Definition 1.1. Let $S = \langle \Omega, F \rangle$ be a relational schema, where

$\Omega = \{ A_1, A_2, \ldots, A_n \}$ is the set of attributes,

$F = \{ L_i \longrightarrow R_i \mid i=1,2,\ldots,k;\ L_i,\ R_i \subseteq \Omega \}$

is the set of functional dependencies, and $Z \subseteq \Omega$, be an arbitrary subset of Ω. We define a new relational schema $\langle \Omega_1, F_1 \rangle$ by:

$$\Omega_1 = \Omega \setminus Z$$

$$F_1 = \{ L_i \setminus Z \rightarrow R_i \setminus Z \mid (L_i \rightarrow R_i) \in F,\quad i=1,\ldots,k \}$$

Then $\langle \Omega_1, F_1 \rangle$ is said to be obtained from $\langle \Omega, F \rangle$ by a Z--translation, and the notation

$$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z$$

is used.

Remarks

1) Depending on the characteristic properties of the class chosen, the corresponding class of translations has its own characteristic features.

2) With the Z-translation just defined above, a functional dependency of type $\emptyset \rightarrow Y$ may occur in $\langle \Omega_1, F_1 \rangle$ that has no ordinary semantic but carries information from the old relational schema to the new one.

In particular, the possibility that $\emptyset$ turns out to be a key of $\langle \Omega_1, F_1 \rangle$ is not excluded.

The next lemma is fundamental for the paper.

Lemma 1.1. Let $\langle \Omega, F \rangle$ be a relational schema and $\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z$, $Z \subseteq \Omega$,

then

a) $X \xrightarrow[F]{} Y$ implies $X\setminus Z \xrightarrow[F_1]{} Y\setminus Z$

b) $X \xrightarrow[F_1]{} Y$ implies $X \cup Z \xrightarrow[F]{} Y \cup Z$

where $X \xrightarrow[F]{} Y$ means $(X \to Y) \in F^+$ and similarly, $X \xrightarrow[F_1]{} Y$ for $(X \to Y) \in F_1^+$.

Proof.

For the part a) of the lemma, we shall prove that

$$X_F^+ \setminus Z \subseteq (X \setminus Z)^+_{F_1} \qquad (1)$$

By the algorithm for finding the closure X^+ of X in [2] with $X_F^{(o)} = X$, $(X\setminus Z)_F^{(o)} = X \setminus Z$ we have

$$X_F^{(o)} \setminus Z \subseteq (X \setminus Z)_{F_1}^{(o)}$$

Supposing that (1) holds for i, that is

$$X_F^{(i)} \setminus Z \subseteq (X \setminus Z)_{F_1}^{(i)}, \qquad (2)$$

we prove that (1) holds for (i+1) as well.
Indeed we have

$$X_F^{(i+1)} \setminus Z = (X_F^{(i)} \cup (\bigcup_{\substack{L_J \subseteq X_F^{(i)} \\ (L_J \to R_J) \in F}} R_J)) \setminus Z =$$

$$(X_F^{(i)} \setminus Z) \cup (\bigcup_{\substack{L_J \subseteq X_F^{(i)} \\ (L_J \to R_J) \in F}} R_J \setminus Z) \qquad \subseteq$$

$$\subseteq \quad (X\setminus Z)^{(i)}_{F_i} \cup \Big(\bigcup_{L_J \subseteq X_F^{(i)}} (R_J \setminus Z)\Big)$$

(by virtue of the inductive assumption (2)).

On the other hand, from $L_J \subseteq X_F^{(i)}$ and the inductive assumption (2), we have:

$$L_J \setminus Z \subseteq X_F^{(i)} \setminus Z \subseteq (X\setminus Z)^{(i)}_{F_1}$$

Consequently:

$$X_F^{(i+1)} \setminus Z \subseteq (X\setminus Z)^{(i)}_{F_1} \cup \Big(\bigcup_{L_J \subseteq X_F^{(i)}} (R_J\setminus Z)\Big) \subseteq (X \setminus F)^{(i+1)}_{F_1}$$

Thus (1) has been proved.
Now, it is well known that

$$X \xrightarrow[F]{} Y \Leftrightarrow Y \subseteq X_F^+$$

Hence, from $X \xrightarrow[F]{} Y$, we have:

$$Y \setminus Z \subseteq X_F^+ \setminus Z \subseteq (X\setminus Z)^+_{F_1}$$

That is,

$$X\setminus Z \xrightarrow[F_1]{} Y\setminus Z$$

Similarly, for the part b) of the lemma, we shall prove by induction that

$$X^+_{F_1} \cup Z \subseteq (X \cup Z)^+_F \; . \tag{3}$$

By the algorithm for finding the closure X^+ of X we have

$$X_{F_1}^{(o)} \cup Z \subseteq (X \cup Z)_F^{(o)}.$$

Supposing that (3) holds with (i), that is

$$X_{F_1}^{(i)} \cup Z \subseteq (X \cup Z)_F^{(i)}, \tag{4}$$

we shall prove that (3) also holds for (i+1).

Indeed we have: $X_{F_1}^{(i+1)} \cup Z = X_{F_1}^{(i)} \cup (\bigcup_{L_J \setminus Z \subseteq X_{F_1}^{(i)}} (R_J \setminus Z)) \cup Z =$

$$(X_{F_1}^{(i)} \cup Z) \cup (\bigcup_{L_J \setminus Z \subseteq X_{F_1}^{(i)}} (R_J \setminus Z)) \subseteq (X \cup Z)_F^{(i)} \cup (\bigcup_{L_J \setminus Z \subseteq X_{F_1}^{(i)}} R_J)$$

(by the inductive assumption (4)).

On the other hand, from $L_J \setminus Z \subseteq X_{F_1}^{(i)}$ and (4) we have

$$L_J \subseteq X_{F_1}^{(i)} \cup Z \subseteq (X \cup Z)_F^{(i)}$$

Consequently:

$$X_{F_1}^{(i+1)} \cup Z \subseteq (X \cup Z)_F^{(i)} \cup (\bigcup_{L_J \setminus Z \subseteq X_{F_1}^{(i)}} R_J) \subseteq (X \cup Z)_F^{(i+1)}$$

Thus (3) has been proved.

From $X \xrightarrow[F_1]{} Y$ we have $Y \subseteq X_{F_1}^+$ hence

$$Y \cup Z \subseteq X_{F_2}^+ \cup Z \subseteq (X \cup Z)_F^+$$

showing that: $X \cup Z \xrightarrow[F]{} Y \cup Z$

The proof is complete.

Definition 1.2.

Let $S = \langle \Omega, F \rangle$ be a relational schema. Let $\mathcal{K}(\Omega, F)$ be the set of all keys of S and

$$H = \bigcup_{X_i \in \mathcal{K}(\Omega,F)} X_i, \quad G = \bigcap_{X_i \in \mathcal{K}(\Omega,F)} X_i$$

Now, we give a classification of the relational schemas as follows:

$$\mathcal{L}_o = \{ \langle \Omega, F \rangle \mid \langle \Omega, F \rangle \text{ is a relational schema} \}$$

$$\mathcal{L}_1 = \{ \langle \Omega, F \rangle \in \mathcal{L}_o \mid \Omega = L \cup R \}$$

$$\mathcal{L}_2 = \{ \langle \Omega, F \rangle \in \mathcal{L}_o \mid L \subseteq R = \Omega \}$$

$$\mathcal{L}_3 = \{ \langle \Omega, F \rangle \in \mathcal{L}_o \mid R \subseteq L = \Omega \}$$

$$\mathcal{L}_4 = \{ \langle \Omega, F \rangle \in \mathcal{L}_o \mid L = R = \Omega \}$$

From the above classification, it is easily seen that:

α) $\mathcal{L}_4 \subseteq \mathcal{L}_3 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_o$

β) $\mathcal{L}_4 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_o$

γ) $\mathcal{L}_4 = \mathcal{L}_2 \cap \mathcal{L}_3$

Figure 1 shows the hierarchy of classes $\mathcal{L}_o, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$.

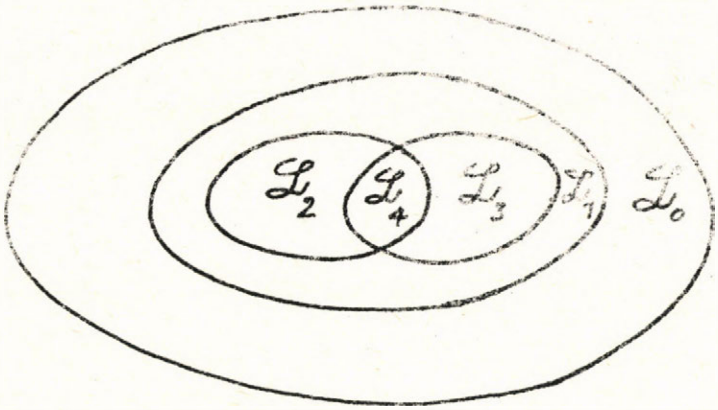


Fig. 1.

We are now in a position to prove the following theorems.

Theorem 1.1. Let $\langle \Omega, F \rangle$ be a relational schema, $Z \subseteq G$ $\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z$. Then X is a key of $\langle \Omega_1, F_1 \rangle$ iff $X \cap Z = \emptyset$ and $X \cup Z$ is a key of $\langle \Omega, F \rangle$.

Proof.

We first prove the necessity. Suppose that X is a key of $\langle \Omega_1, F_1 \rangle$. Obviously $X \subseteq \Omega_1$, therefore $X \cap Z = \emptyset$. Since X is a key of $\langle \Omega_1, F_1 \rangle$, $X \xrightarrow[F_1]{} \Omega_1$. Taking lemma 1.1. into account we get

$$X \cup Z \xrightarrow[F]{} \Omega_1 \cup Z = \Omega,$$

showing that $X \cup Z$ is a superkey of $\langle \Omega, F \rangle$. Were $X \cup Z$ not a key of $\langle \Omega, F \rangle$ then there would exist a key $\bar{X}$ of $\langle \Omega, F \rangle$ such that

$$Z \subseteq \bar{X} \subset X \cup Z.$$

Consequently, there would exist an $X_1 \subset X$ such that

$$\bar{X} = X_1 \cup Z, \quad X_1 \cap Z = \emptyset$$

Since $\bar{X}$ is supposed to be a key of $\langle \Omega, F \rangle$, $X_1 \cup Z \xrightarrow[F]{} \Omega$.

Applying lemma 1.1, clearly

$$(X_1 \cup Z) \setminus Z \xrightarrow[F_1]{} \Omega \setminus Z,$$

that is $\qquad X_1 \xrightarrow[F_1]{} \Omega_1 .$

This contradicts the hypothesis that $\bar{X}$ is a key of $\langle \Omega_1, F_1 \rangle$. Thus $X \cup Z$ is a key of $\langle \Omega, F \rangle$.

We now turn to the proof of sufficiency. Suppose that $X \cap Z = \emptyset$ and $X \cup Z$ is a key of $\langle \Omega, F \rangle$. We have to show that X is a key of $\langle \Omega_1, F_1 \rangle$.

Since $X \cup Z$ is a key of $\langle \Omega, F \rangle$ we have

$$X \cup Z \xrightarrow[F]{} \Omega .$$

By virtue of lemma 1.1, we get

$$(X \cup Z) \setminus Z \xrightarrow[F_1]{} \Omega \setminus Z .$$

Consequently (from $X \cap Z = \emptyset$):

$$X \xrightarrow[F_1]{} \Omega_1 ,$$

showing that X is a superkey of $\langle \Omega_1, F_1 \rangle$. Assume that X is not a key of $\langle \Omega_1, F_1 \rangle$. Then, there would exist a key $\bar{X}$ of $\langle \Omega_1, F_1 \rangle$ such that

$$\bar{X} \subset X \quad \text{and} \quad \bar{X} \xrightarrow[F_1]{} \Omega_1 .$$

Applying lemma 1.1, it follows:

$$\bar{X} \cup Z \xrightarrow[F]{} \Omega_1 \cup Z = \Omega ,$$

where $\bar{X} \cup Z \subset X \cup Z$.

This contradicts the fact that $X \cup Z$ is a key of $\langle \Omega, F \rangle$
Hence X is a key of $\langle \Omega_1, F_1 \rangle$.

The proof is complete.

Theorem 1.2.

Let $\langle \Omega, F \rangle$ is a relational schema, $Z \subseteq \Omega$, $Z \cap H = \emptyset$ and $\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z$.

Then X is a key of $\langle \Omega_1, F_1 \rangle$ iff X is a key of $\langle \Omega, F \rangle$.

Proof.

(i) (The necessity)

Suppose that X is a key of $\langle \Omega_1, F_1 \rangle$. Obviously $X \xrightarrow[F_1]{} \Omega_1$.
By virtue of lemma 1.1, we have

$$X \cup Z \xrightarrow[F]{} \Omega_1 \cup Z = \Omega \quad ,$$

showing that $X \cup Z$ is a superkey of $\langle \Omega, F \rangle$. Hence, there exists a key $\bar{X}$ of $\langle \Omega, F \rangle$ such that $\bar{X} \subseteq X \cup Z$.
Since $Z \cap H = \emptyset$ then $\bar{X} \cap Z = \emptyset$. From this, it is easy to see that $\bar{X} \subseteq X$. There are two possible cases:

a) $\bar{X} = X$ Then obviously X is a key of $\langle \Omega, F \rangle$.

b) $\bar{X} \subset X$ Since $\bar{X}$ is a key of $\langle \Omega, F \rangle$, $\bar{X} \xrightarrow[F]{} \Omega$.

Applying lemma 1.1., we have

$$\bar{X} \setminus Z \xrightarrow[F_1]{} \Omega \setminus Z,$$

that is $$\bar{X} \xrightarrow[F_1]{} \Omega_1.$$

This contradicts the fact that X is a a key of $\langle \Omega_1, F_1 \rangle$

(ii) (The sufficiency)

Suppose that X is a key of $\langle \Omega, F \rangle$. We have to prove that X is also a key of $\langle \Omega_1, F_1 \rangle$. We have, by the definition of keys

$$X \xrightarrow[F]{} \Omega .$$

Applying lemma 1.1:

$$X \setminus Z \xrightarrow[F_1]{} \Omega \setminus Z = \Omega_1 .$$

Since $Z \cap H = \emptyset$, it follows $X \cap Z = \emptyset$. Consequently,

$$X \xrightarrow[F_1]{} \Omega_1$$

showing that X is a superkey of $\langle \Omega_1, F_1 \rangle$.

Now, assume the contrary that X is not a key of $\langle \Omega_1, F_1 \rangle$. Then there would exist a key $\bar{X}$ of $\langle \Omega_1, F_1 \rangle$ such that $\bar{X} \subset X$. Obviously

$$\bar{X} \xrightarrow[F_1]{} \Omega_1 .$$

We invoke lemma 1.1. to deduce

$$\bar{X} \cup Z \xrightarrow[F]{} \Omega_1 \cup Z = \Omega ,$$

showing that $\bar{X} \cup Z$ is a superkey of $\langle \Omega, F \rangle$. Consequently, there exists a key $\bar{\bar{X}}$ of $\langle \Omega, F \rangle$ such that

$$\bar{\bar{X}} \subseteq \bar{X} \cup Z, \quad \bar{\bar{X}} \cap Z = \emptyset .$$

From this $\bar{\bar{X}} \subseteq \bar{X} \subset X.$

This contradicts the hypothesis that X is a key of $\langle \Omega, F \rangle$.

The proof is complete.

Based on theorems 1.1 and 1.2, in the following we investigate only the class of Z - translations with $Z\neq\emptyset$, $Z=Z_1\cup Z_2$, $Z_1\cap Z_2=\emptyset$. $Z_1\subset G$, $Z_2\cap H=\emptyset$.
Bearing this in mind, if

$$<\Omega_1,F_1> = <\Omega,F> - Z$$

then applying theorem 1.2 and 1.1 one after another to the Z_2-translation and the Z_1-translation, we have: X is a key of $<\Omega_1,F_1>$ if and only if $X\cap Z=\emptyset$ and $X\cup Z_1$ is a key of $<\Omega,F>$. For the sake of convenience, we use in the sequel the notation

$$<\Omega,F> \xRightarrow[\rho=(Z_1,Z_1)]{} <\Omega_1,F_1>$$

where the meaning of ρ is obvious.
To continue, let us recall a result in [1]. Let $S=<\Omega,F>$ be a relationsl schema, where

$$\Omega = \{A_1,\dots,A_n\} \text{ - the set of attributes,}$$

$$F = \{L_i\to R_i \mid L_i,R_i\subset\Omega,\ \ i=1,\dots,k\} \text{ - the set of}$$

functional dependencies.
Let us denote

$$L = \bigcup_{i=1}^{k} L_i,\ R = \bigcup_{i=1}^{k} R_i$$

Then, the necessary condition for which X is a key of S is that

$$\Omega\setminus R \subseteq X \subseteq (\Omega\setminus R)\cup(L\cap R)$$

For $V\subseteq\Omega$ we denote $\bar{V}=\Omega\ V$. It is easily seen that

$$\overline{L\cup R} \subseteq \Omega\setminus R \subseteq G$$

$$L\setminus R \subseteq \Omega\setminus R \subseteq G$$

$R \setminus L \subseteq \bar{H}$, consequently $(R \setminus L) \cap H = \emptyset$, and we have the following lemma:

Lemma 1.2. Let $S=<\Omega,F>$ be a relational schema, $Z \subseteq G$, where G is the intersection of all the keys of S.

Then $(Z^+ \setminus Z) \cap H = \emptyset$,

where H is the union of all the keys of S.

Proof. Assume the contrary that

$$(Z^+ \setminus Z) \cap H \neq \emptyset.$$

Then, there would exist an attribute $A \in Z^+$, $A \bar{\in} Z$ and $A \in H$. Consequently, there exists a key X of $S=<\Omega,F>$ such that $A \in X$.

Since $A \in Z^+$ and $A \bar{\in} Z$ we infer that $Z \subseteq X \setminus A$.

Hence

$$X \setminus A \overset{*}{\to} Z \overset{*}{\to} Z^+ \overset{*}{\to} A$$

with $A \in X$

This contradicts to the fact that X is a key of S.

The proof is complete.

From the results mentioned just above the following theorems are obvious.

Theorem 1.3. Let $S=<\Omega,F>$ be a relational schema belonging to $\mathcal{L}_0$,

$$<\Omega_1,F_1> = <\Omega,F> - \overline{L \cup R}.$$

Then

$$<\Omega,F> \xRightarrow[\rho=(\overline{L\cup R},\overline{L\cup R})]{} <\Omega_1,F_1>$$

with

$$<\Omega_1,F_1> \in \mathcal{L}_1.$$

Proof. As remarked above $\overline{L \cup R} \subseteq G$

Applying Theorem 1.1. to the Z-translation with $Z=\overline{L \cup R}$ we have

$$<\Omega,F> \xrightarrow[\rho=(\overline{L\cup R},\overline{L\cup R})]{} <\Omega_1,F_1>$$

The theorem 1.3 is illustrated by Figure 2.

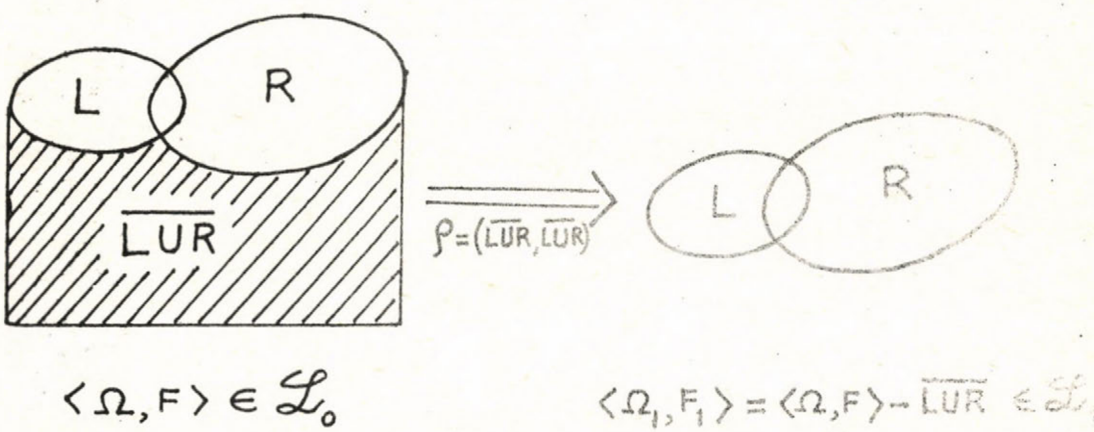


Fig. 2.

Example 1. Let there be given $S = <\Omega,F>$

with $\Omega = \{a,b,c,d,e\}$, $F = \{c\to d,\ d\to e\}$.

We have $\overline{L \cup R} = ab$.

Consider $<\Omega_1,F_1> = <\Omega,F> - ab$.

Obviously $\Omega_1 = \{c,d,e\}$, $F_1 = \{c\to d,\ d\to e\}$.

It is easily seen that c is the unique key of $<\Omega_1,F_1>$, hence abc is the unique key of $<\Omega,F>$.

Theorem 1.4.

Let $<\Omega,F>$ be a relational schema of $\mathcal{L}_0$,

$$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - (\overline{L \cup R} \cup (L \setminus R)).$$

Then

$$\langle \Omega, F \rangle \xRightarrow[\rho = (\overline{L \cup R} \cup (L \setminus R), \overline{L \cup R} \cup (L \setminus R))]{} \langle \Omega_1, F_1 \rangle$$

with

$$\langle \Omega_1, F_1 \rangle \in \mathcal{L}_2.$$

Proof.

It is clear that

$$Z = \overline{L \cup R} \cup (L \setminus R) = \Omega \setminus R \subseteq G$$

The theorem 1.4 now follows from applying theorem 1.1 to the Z-translation.

Theorem 1.4 is illustrated by figure 3.

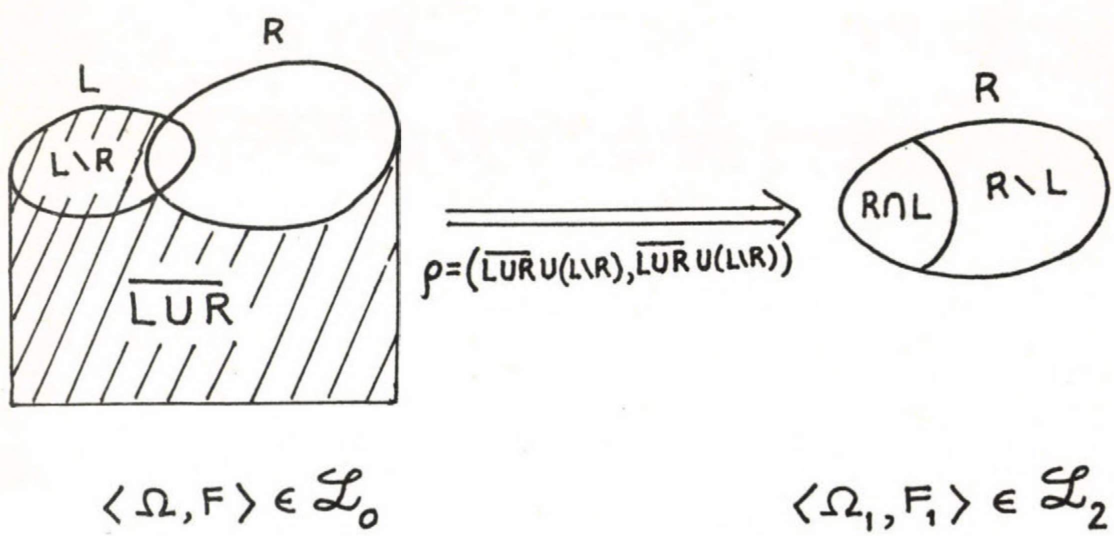


Fig. 3.

Theorem 1.5.

Let $S = <\Omega,F>$ be a relational schema of $\mathcal{L}_o$,

$<\Omega_1,F_1> = <\Omega,F> - (\overline{L\cup R} \cup (R\setminus L))$.

Then

$$<\Omega,F) \xrightarrow[\rho=(\overline{L\cup R}\cup(R\setminus L),\overline{L\cup R})]{} <\Omega_1,F_1>$$

with

$$<\Omega_1,F_1> \in \mathcal{L}_3.$$

Proof. As remarked above, $R\setminus L \subseteq \bar{H}$.

Let $Z = \overline{L\cup R} \cup (R\setminus L) = Z_1 \cup Z_2$,

where $Z_1 = \overline{L\cup R} \subseteq G$, $Z_2 = R\setminus L$, $Z_2 \cap H = \emptyset$-

The theorem 1.5 now follows from sequential applications of theorems 1.2 and 1.1 one after another to the Z_2 - translation and the Z_1 - translation. Theorem 1.5 is illustrated by Fig. 4.

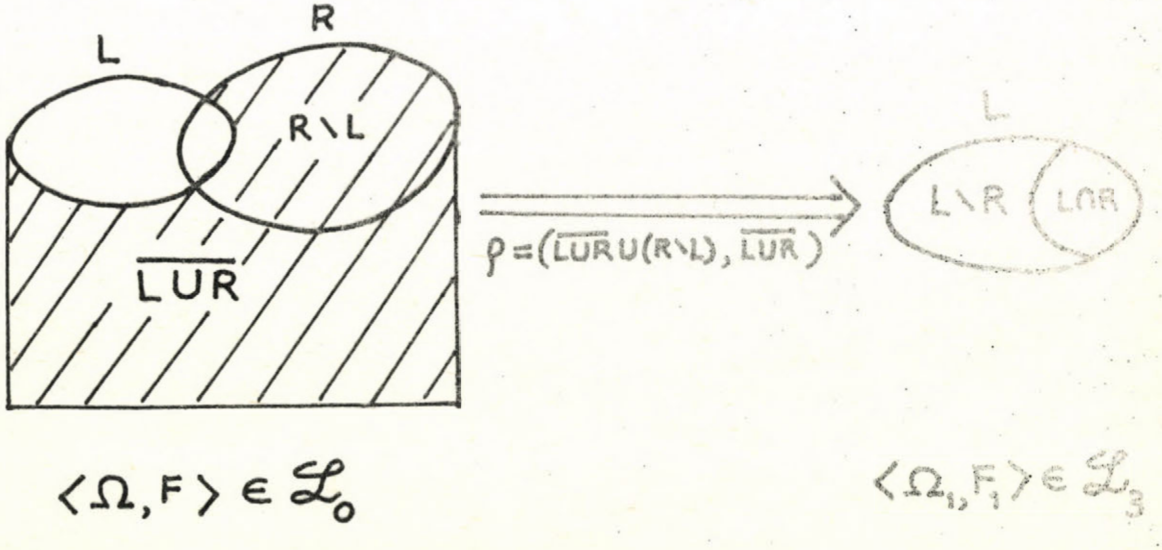


Fig. 4.

<u>*Theorem 1.6.*</u> Let $S = \langle \Omega, F \rangle$ be a relational schema of $\mathcal{L}_o$,

$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - (\overline{L \cup R} \cup (L \setminus R) \cup (R \setminus L))$.

Then

$$\langle \Omega, F \rangle \xRightarrow[=(L\cup R\ \cup (L\ R) \cup (R\ L),\ \overline{L\cup R} \cup (L\ R))]{} \langle \Omega_1, F_1 \rangle,$$

with

$$\langle \Omega_1, F_1 \rangle \in \mathcal{L}_4.$$

<u>*Proof.*</u> Let $Z = \overline{L \cup R}\ \cup (L \setminus R) \cup (R \setminus L) = Z_1 \cup Z_2$,

where $Z_1 = \overline{L \cup R}\ \cup (L \setminus R) = \Omega \setminus R \subseteq G$

$Z_2 = R \setminus L \subseteq \bar{H}$ or equivalently $Z_2 \cap H = \emptyset$

It is obvious that $\langle \Omega_1, F_1 \rangle$ is obtained from $\langle \Omega, F \rangle$ by the Z - translation. The proof of theorem 1.6 is straight-forward. Theorem 1.6 is illustrated by Fig. 5.

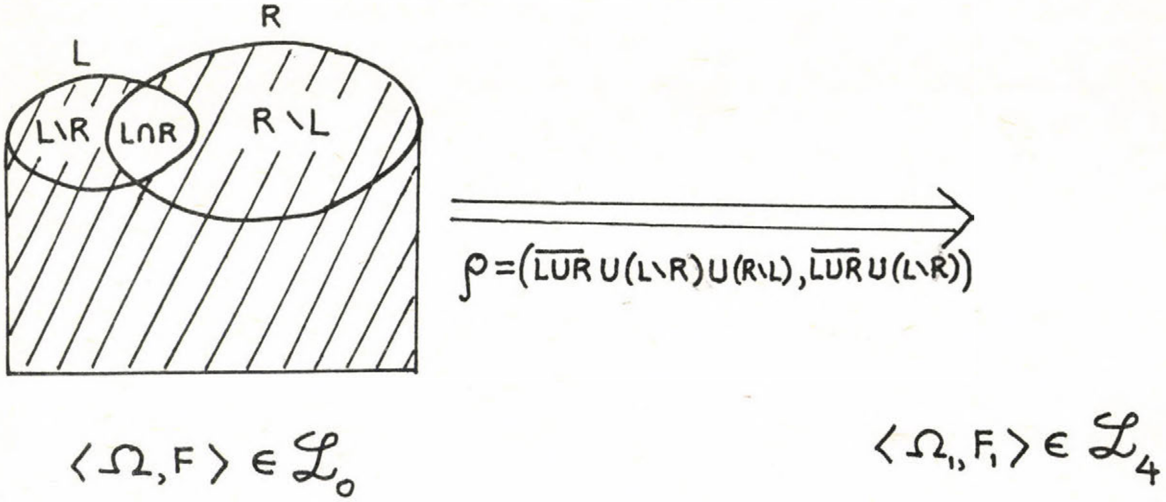


Fig. 5.

Similarly, we can prove the following theorems:

Theorem 1.7.

Let $S = \langle\Omega, F\rangle$ be a relational schema of $\mathcal{L}_1$,

$\langle\Omega_1, F_1\rangle = \langle\Omega, F\rangle - (L \setminus R)$.

Then

$$\langle\Omega, F\rangle \xrightarrow[=(L\ \ R,\ L\ \ R)]{} \langle\Omega_1, F_1\rangle,$$

where $\langle\Omega_1, F_1\rangle \in \mathcal{L}_2$.

Theorem 1.7 is illustrated by Fig. 6.

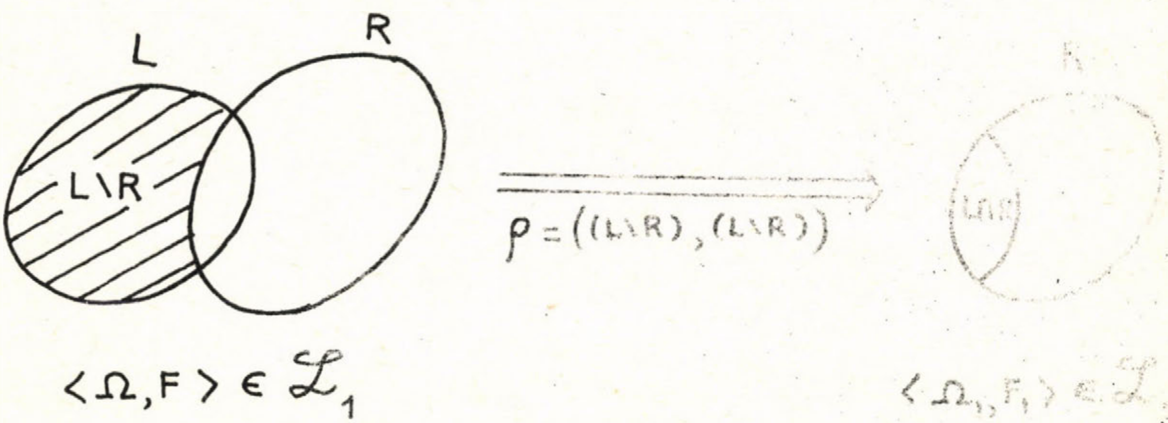


Fig. 6.

Theorem 1.8.

Let $S = \langle\Omega, F\rangle$ be a relational schema of $\mathcal{L}_1$,
$\langle\Omega_1, F_1\rangle = \langle\Omega, F\rangle - (R \setminus L)$.

Then

$$<\Omega,F> \xrightarrow[\rho=(R\setminus L,\ \emptyset)]{} <\Omega_1,F_1>,$$

where

$$<\Omega_1,F_1> \in \mathcal{L}_3.$$

Theorem 1.8. is illustrated by Fig. 7.

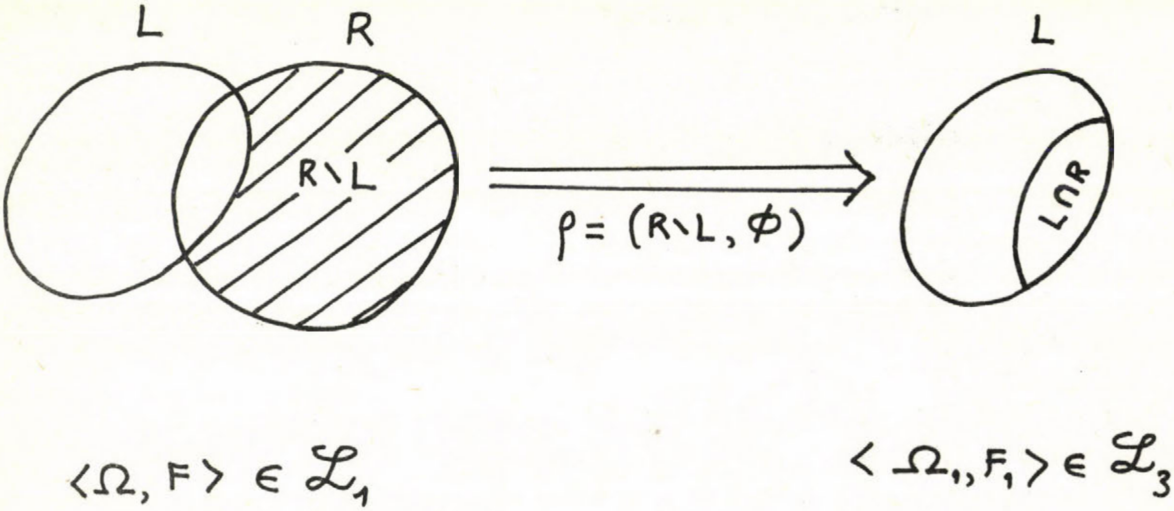


Fig. 7.

Theorem 1.9. Let $S = <\Omega,F>$ be a relational schema of $\mathcal{L}_1$,

$$<\Omega_1,F_1> = <\Omega,F>-((L\setminus R) \cup (R\setminus L)).$$

Then

$$<\Omega,F> \xrightarrow[\rho=((L\setminus R)\cup(R\setminus L),L\setminus R)]{} <\Omega_1,F_1>,$$

where

$$<\Omega_1,F_1> \in \mathcal{L}_4.$$

Theorem 1.9 is illustrated by Fig. 8.

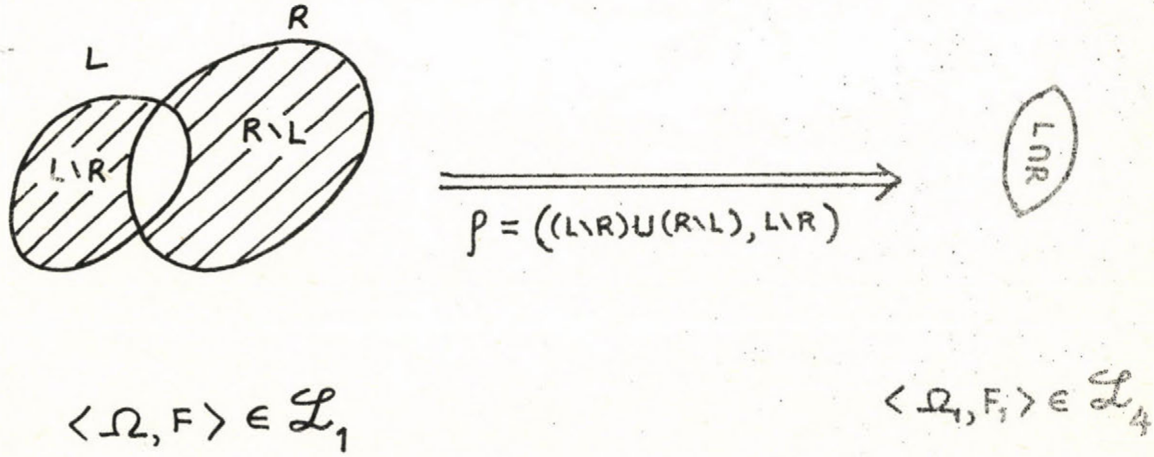


Fig. 8.

<u>*Theorem 1.10.*</u> Let $\langle \Omega, F \rangle$ be a relational schema of $\mathcal{L}_2$,

$$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - (R \setminus L).$$

Then

$$\langle \Omega, F \rangle \xRightarrow[\rho = (R\setminus L, \emptyset)]{} \langle \Omega_1, F_1 \rangle,$$

where

$$\langle \Omega_1, F_1 \rangle \in \mathcal{L}_4$$

Theorem 1.10 is illustrated by Fig. 9.

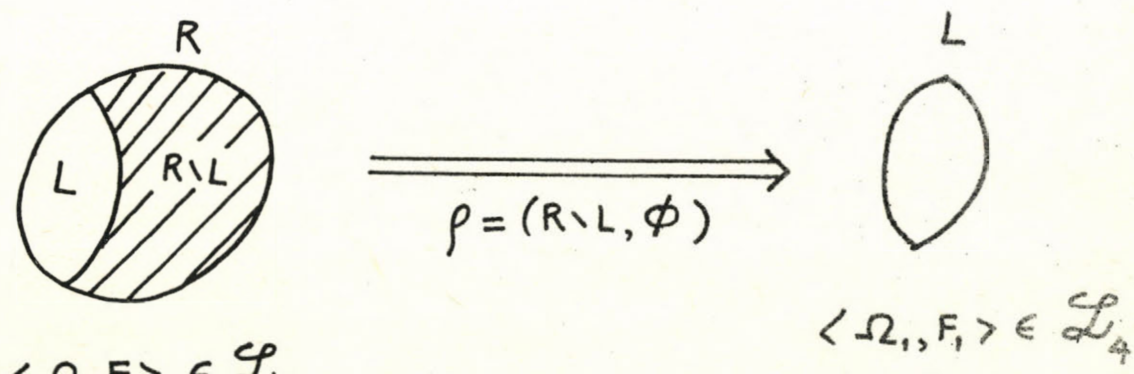


Fig. 9.

Theorem 1.11. Let $<\Omega,F>$ be a relational schema of $\mathcal{L}_3$,

$$<\Omega_1,F_1> = <\Omega,F> - (L \setminus R).$$

Then

$$<\Omega,F> \xRightarrow[=(L\setminus R,L\setminus R)]{} <\Omega_1,F_1>,$$

where $$<\Omega_1,F_1> \in \mathcal{L}_4.$$

Theorem 1.11 is illustrated by Fig. 10

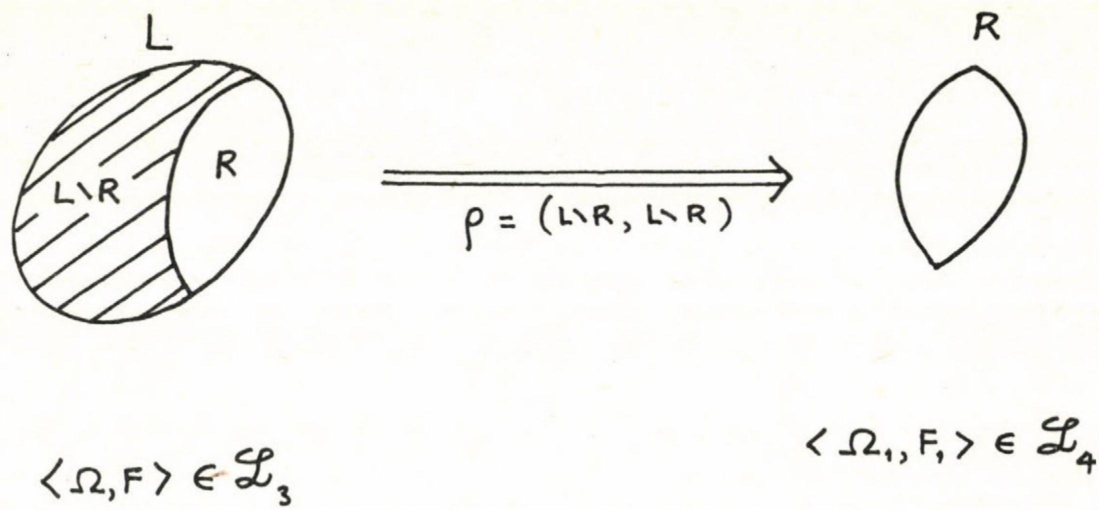


Fig. 10.

Combining theorems 1.3 - 1.11 we have the diagram of translations as illustrated on figure 11.

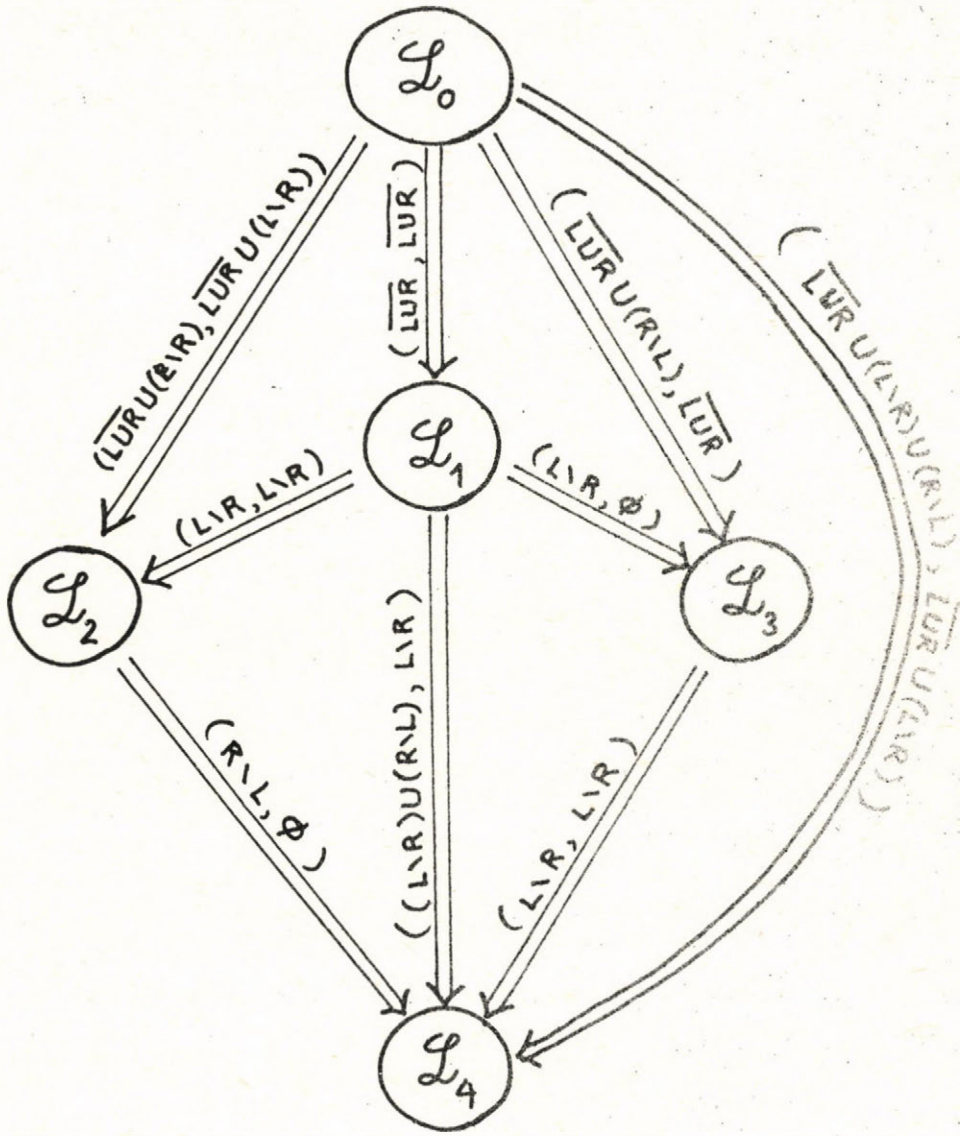


Fig. 11.

Now, the following theorem follows from theorems 1.1, 1.2 and lemma 1.3.

Theorem 1.12.

Let $\langle\Omega,F\rangle$ be a relational of $\mathcal{L}_0$,

$$\langle\Omega_1,F_1\rangle = \langle\Omega,F\rangle - \{\overline{L\cup R} \cup (L\setminus R)^+ \cup (R\setminus L)\}.$$

Then

$$\langle \Omega, F\rangle \xRightarrow[\rho=(\overline{L\cup R}\,\cup\,(L\setminus R)^{+}\,\cup\,(R\setminus L),\ \overline{L\cup R}\,\cup\,(L\setminus R))]{} \langle \Omega_1, F_1\rangle,$$

where

$$\langle \Omega_1, F_1\rangle \in \mathcal{L}_4.$$

Proof.

Put $Z = \overline{L\cup R} \cup (L\setminus R) \cup [(L\setminus R)^{+}\setminus (L\setminus R)] \cup (R\setminus L) = Z_1 \cup Z_2$,

where $Z_1 = \overline{L\cup R} \cup (L\setminus R) = \Omega\setminus R \subseteq G$,

$Z_2 = [(L\setminus R)^{+}\setminus (L\setminus R)] \cup (R\setminus L)$.

Clearly $Z_2 \cap H = \emptyset$.

Applying theorem 1.2 to

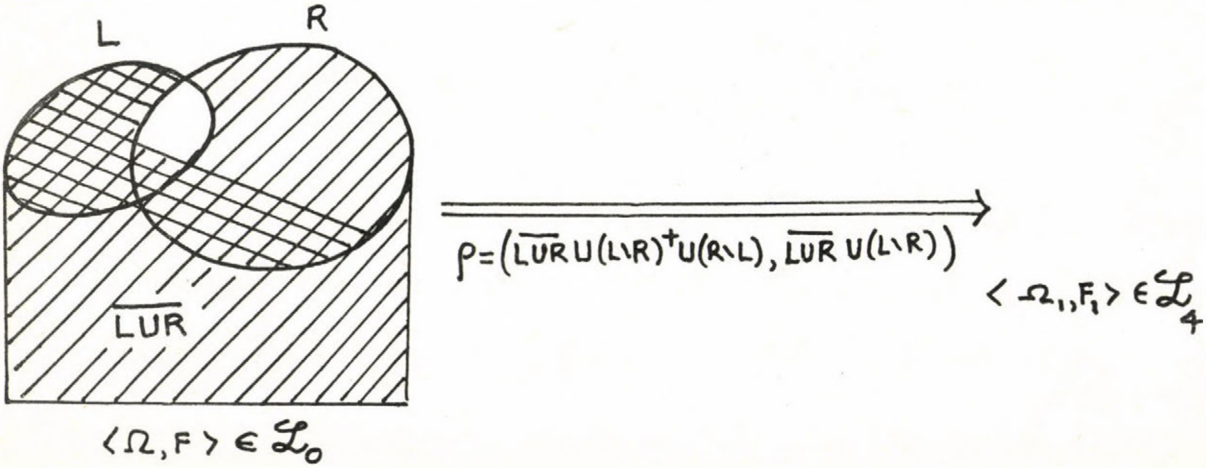
$$\langle \Omega', F'\rangle = \langle \Omega, F\rangle - Z_2,$$

and then, theorem 1.1 to

$$\langle \Omega_1, F_1\rangle = \langle \Omega', F'\rangle - Z_1,$$

the proof of theorem 1.12 is easy.

Theorem 1.12 is illustrated by Fig. 12.



The "double hashing" part is $(L\setminus R)$

Fig. 12.

From the just mentioned results, we have the following diagram of translations of relational schemas (Fig. 13).

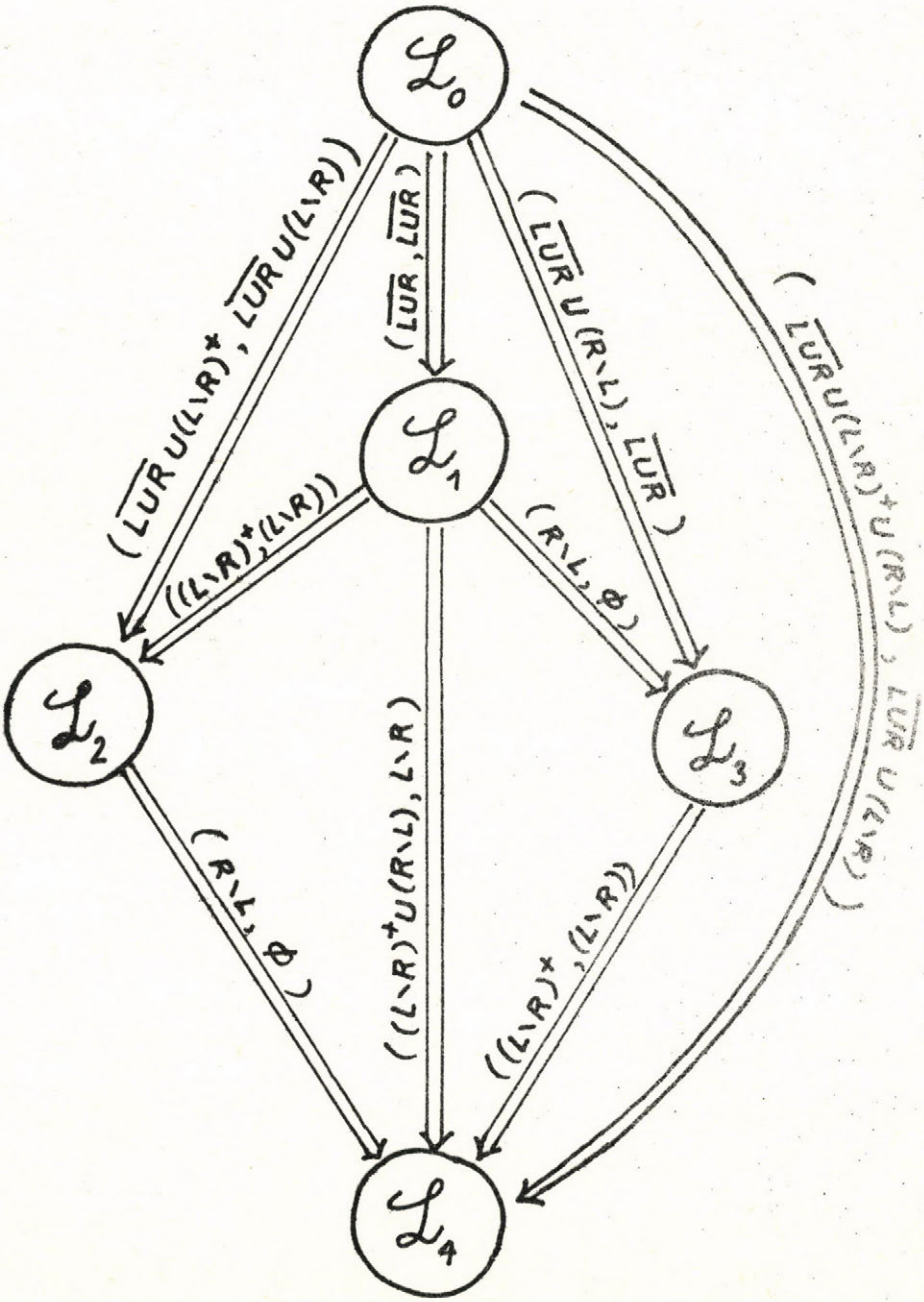


Fig. 13.

<u>*Example 2*</u>. Let Ω = a b h g q m n v w k l,

F = {a→b, b→h, g→q, kv→w, w→vl}.

we have

L = abgkvw; R = bhqwvl; R\L = hql;

$L \setminus R = kga$; $(L \setminus R)^+ = kgabhq$; $\overline{L \cup R} = mn$;

$(R \setminus L) \cup (L \setminus R)^+ \cup (\overline{L \cup R}) = mnkgabhql$

$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - mnkgabhql = \langle wv, \{v \to w, \ w \to v\} \rangle$.

It is easily seen that v and w are keys of $\langle \Omega_1, F_1 \rangle$. On the other hand

$$(\overline{L \cup R}) \cup (L \setminus R) = m\ n\ k\ g\ a$$

Consequently $mnkgav$ and $mnkgaw$ are keys of $\langle \Omega, F \rangle$.

§2.

In this section we investigate some properties of the so-called nontranslatable relational schemas.

Definition 2.1. Let $S = \langle \Omega, F \rangle$ be a relational schema. S is called translatable if and only if there exist certain sets $Z_1, Z_2 \subseteq \Omega$ such that:

(i) $Z_1 \neq \emptyset$

(ii) X is a key of $\langle \Omega_1, F_1 \rangle$ iff $X \cap Z_2 = \emptyset$ and $X \cup Z_2$ is a key of $\langle \Omega, F \rangle$, where $\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z_1$.

Otherwise S is called nontranslatable.

Theorem 2.1. Let $S = \langle \Omega, F \rangle$ be a translatable relational schema with Z_1, Z_2 as defined above.

Then

$$H \setminus G = H_1 \setminus G_1,$$

where H and G (and similarly H_1 and G_1) are defined in definition 1.2.

Proof.

$$\text{Let } \langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - Z_1.$$

Since X is a key of $\langle \Omega_1, F_1 \rangle$ iff $X \cap Z_2 = \emptyset$ and $X \cup Z_2$ is a key of $\langle \Omega, F \rangle$, it follows:

$$H = H_1 \cup Z_2, \quad Z_2 \cap H_1 = \emptyset,$$
$$G = G_1 \cup Z_2, \quad Z_2 \cap G_1 = \emptyset, \quad .$$

hence

$$H \setminus G = (H_1 \cup Z_2) \setminus (G_1 \cup Z_2) = ((H_1 \cup Z_2) \setminus Z_2) \setminus G_1 = H_1 \setminus G_1$$

(because $Z_2 \cap H_1 = \emptyset$).

Combining theorems 1.1, 1.2 with theorem 2.1, the following theorem is obvious:

Theorem 2.2. Let $S = \langle \Omega, F \rangle$ be a relational schema. $\langle \Omega, F \rangle$ is non translatable iff $H = \Omega$ and $G = \emptyset$.

Theorem 2.3. Let $S = \langle \Omega, F \rangle$ be a relational schema,

$$\langle \Omega_1, F_1 \rangle = \langle \Omega, F \rangle - (G \setminus \bar{H})$$

Then:

a) $\langle \Omega, F \rangle \xRightarrow[\rho = (G \cup \bar{H}, G)]{} \langle \Omega_1, F_1 \rangle.$

b) $\langle \Omega_1, F_1 \rangle$ is non translatable.

c) $\langle \Omega_1, F_1 \rangle \in \mathcal{L}_4.$

Proof. Let $Z = G \cup \bar{H} = Z_1 \cup Z_2,$

where $Z_1 = G \subseteq G$, $Z_2 = \bar{H}$ (clearly $Z_2 \cap H = \emptyset$).

Hence part a) of the theorem is obvious. To prove b), we have only to show that

$$G_1 = \emptyset \text{ and } H_1 = \Omega_1.$$

From a) it is clear that X is a key of $\langle\Omega_1, F_1\rangle$ iff $X \cap G = \emptyset$ and $X \cup G$ is a key of $\langle\Omega, F\rangle$.

Therefore, $G = G \cup G_1$, $G \cap G_1 = \emptyset$
$H = G \cup H_1$, $G \cap H_2 = \emptyset$-

Hence

$$G_1 = G \setminus G = \emptyset$$

and $H_1 = H \setminus G$.

On the otherhand we have

$$\Omega_1 = \Omega \setminus (G \cup \bar{H}) = (\Omega \setminus \bar{H}) \setminus G = H \setminus G = H_1.$$

To prove c) we have to show that

$$L^1 = R^1 = \Omega_1$$

where L^1 and R^1 are the union of all the left sides and right sides of all functional dependencies of F_1, respectively.

It is known [1] that

$$\Omega_1 \setminus R^1 \subseteq G_1 = \emptyset.$$

On the other hand

$$R^1 \subseteq \Omega_1.$$

Hence $R^1 = \Omega_1$.

There remained to prove $L^1 = \Omega_1$.
Were this false, there would exist an $A \in \Omega_1 \setminus L^1$
Since $R^1 = \Omega_1$, we have

$$A \in R^1 \text{ and } A \bar{\in} L^1.$$

From $\Omega_1 = H_1$ there exists a key X of $<\Omega_1, F_1>$ such that

$$A \in X \text{ amd } X \overset{*}{\to} \Omega_1$$

Since $A \bar{\in} L^1$ it follows from [1] that

$$X \setminus A \overset{*}{\to} \Omega_1 \setminus A.$$

Evidently

$$L^1 \subseteq \Omega_1 \setminus A$$

and from this,

$$X \setminus A \overset{*}{\to} \Omega_1 \setminus A \overset{*}{\to} L \overset{*}{\nrightarrow} R^1 \overset{*}{\to} A.$$

This contradicts the fact that X is a key of $<\Omega_1, F_2>$, hence $L^1 = \Omega_1$.

The proof is complete.

From the proof of c) we conclude that all non translatable relational schemeas are of type $\mathcal{L}_4$.

Theorem 2.4. Let $S = <\Omega, F>$ be a relational schema from $\mathcal{L}_4$ satisfying the following conditions:

(i) $L_i \cap R_i = \emptyset \quad \forall i = 1,2,\ldots,k,$

(ii) for each L_i, $i=1,\ldots,k$ there exists a key X_i such that $L_i \subseteq X_i$.

Then $<\Omega\ F>$ is a nontranslatable relational schema.

Proof. We have to prove that $H = \Omega$ and $G = \emptyset$-
In fact, from $<\Omega, F> \in \mathcal{L}_4$ we have $L = R = \Omega$.
By virtue of the hypothesis of the theorem we have

$$\Omega = L = \bigcup_{i=1}^{k} L_i \subseteq \bigcup_{i=1}^{k} X_i \subseteq H \subseteq \Omega$$

Consequently, $H = \Omega$.

To prove $G = \emptyset$ we first show that if $L_i \to R_i$ and X_i is a key such that $L_i \subseteq X_i$ then $X_i \cap R_i = \emptyset$.

Assume the contrary that $X_i \cap R_i \neq \emptyset$.

Then, there would exist an $A \in X_i \cap R_i$.

Since $L_1 \cap R_i = \emptyset$ clearly $A \bar{\in} L_i$. Therefore $L_i \subseteq X_i \setminus A$.

On the other hand

$$X_i \setminus A \overset{*}{\to} L_i \overset{*}{\to} R_i \overset{*}{\to} A,$$

showing that X is not a key of $<\Omega,F>$. We thus arrive at a contradiction. From $X_i \cap R_i = \emptyset$, it follows:

$$X_i \subseteq \Omega \setminus R_i .$$

Thus

$$G \subseteq \bigcap_{i=1}^{k} X_i \subseteq \bigcap_{i=1}^{k} (\Omega \setminus R_i) = \Omega \setminus \bigcup_{i=1}^{k} R_i .$$

Since $R = \Omega$ clearly

$$G \subseteq \Omega \setminus \Omega = \emptyset .$$

showing that $G = \emptyset$.

The proof is complete.

ACKNOWLEDGEMENT

The authors would like to take this opportunity to express their sincere thanks to Prof. Dr. J. Demetrovics and Dr. A. Békéssy for their valuable comments and suggestions.

REFERENCES

[1] Ho Thuan and Le van Bao. Some results about keys of relational schemas (to appear)

[2] Ullman, J. Principles of database systems, Prentice Hall, 1980.

ÖSSZEFOGLALÁS

A RELÁCIÓS SÉMÁK ELTOLÁSAI

Ho Thuan és *Le Van Bao*

A cikkben a szerzők bevezetik a relációs sémák eltolásainak fogalmát. Elindulva az adott sémából eltolással általában egyszerübb sémák nyerhetők.

A szerzők a következő kérdésekkel foglalkoznak:

- a relációs sémák osztályozása az eltolhatóság szempontjából;
- az eltolások bizonyos osztályainak tulajdonságai;
- u.n. nem eltolható sémák tulajdonságai.

ТРАНСЛЯЦИИ РЕЛЯЦИОННЫХ СХЕМ

В статье вводится понятие трансляции реляционных схем и изучаются основные вопросы, такие как:

- классификация схем с точки зрения их трансляций;
- свойства некоторых классов трансляций;
- свойства схем, которые не позволяют трансляций.

Közlemények 30/1984 37-51

SOME REMARKS ON STATISTICAL DATA PROCESSING

J. DEMETROVICS, P. KERÉKFY, A. KRÁMLI, M. RUDA
Computer and Automation Institute
Hungarian Academy of Sciences

1. INTRODUCTION

The so-called "software problem" or "software crisis" is the most important matter at issue in computer science. Several papers are devoted to discuss different aspects of the crisis (see e.g. [9]). There is a lot of contributions both in the theory and the practice that aims to resolve parts of the problems. These efforts can be classified into three major categories: very high level languages (VHLL's), logic-based and knowledge-based systems. The VHLL's continue the historical evolutionary trend of software development by developing new programming languages in which the program can be described at a higher level of abstraction. Statistical data processing system requires numerous new concepts and methods [21]. These tasks give rise to difficulties mainly in large and complicated statistical investigations. With our work started some years ago we wished to obtain results just in this field.

Our first attempts in this field were concluded from the Hungarian Hospital Morbidity Study [3] producing a simple statistical information system. We intended to answer very quickly questions about a large mass of data. One of our tools was collecting a large variety of statistical data in advance (producing the so-called "table files"). A very fast information retrieval can be realized using the collected data, its speed does not depend on the size of the sample.

Another basic technical tool was program generating [15,21]. In this way we built up system SIS77 (Statistical Information System 1977) for the Hungarian Hospital Morbidity Study [3] that acted very well in COMECON cooperation [15].

System GENERA [12] was developed for the extension and wide-ranging applications of the method used in SIS77. It gives assistance to a program generator technique that makes the programming and usage of optimal-performance procedures possible.

GENERA processes directives imbedded in a host language, so it is quite flexible and can easily be extended or modified. Any supplementary program can be written in the host language. The generator procedures are independent modules written in high-level languages so they are easy to survey and correct. Error detection is supported by the simple and standardized structures of generated program fragments. On the contrary, traditional advanced programming tools (high level languages, program packages) are usually closed, the user cannot modify or supplement them. Some additions are allowed (as e.g. in BMDP) but they does not touch the inner structure of the system. It must be mentioned as an advantage of these advanced software tools that the user need not have much knowledge on the computer background. But it can make the usage mysterious. The user becomes "alienated" from the system, from the compute science embodied in it and from the applied program and the results. Consequently the user may be unable to give preliminary estimations on the computer resources needed, it provides means for maneuvers that cannot be taken in or controlled, and the user may not be able to interpret the results correctly. On the other hand, the user unfamiliar with the system can not necessarily use it properly even if the rules are simple when he does not know the mathematical, software and organizational background. Observing an error in the system, it cannot be located and corrected by simple means.

Utilizing GENERA the user organizes his work in the host language (a high-level language) and it can be expected that he has a good insight into the program. Besides, the directives of a system managed by GENERA (such as SIS79) release the user from the most cumbersome work in programming, moreover, its macro processor makes the production of parameter-controlled programs possible. Compared to the traditional methods, the generator procedure rather than the running program receives the parameters. This is important to efficiency. However it should be noted that GENERA is not a macro generator in the traditional meaning of the word.

Coming back to the matter of statistical data processing, we shall touch upon some other problems.

While processing large and complicated data sets, beyond the problem of selecting proper software tools interesting mathematical (optimization) problems arise. They can emerge while designing the codes used, sampling, designing the processes and data storage.

Data checking, transformation and, in general, analysis of functions providing control, transformation or selection of the sample form another group of important questions. These tasks require (in the case of a large and complicated system) modelling of strange functions and convenient description of large code tables.

FORTRAN is used very often to write programs for statistical data processing, some well-known systems (such as BMDP or SPSS) utilize it. Present implementations of GENERA have FORTRAN as an optional host language (beside PL/1). Data description and I/O procedures of FORTRAN are sometimes inconvenient and slow. This fact inspired us to work out some procedures for input--output, data description and storage in systems SIS77 and SIS79.

2. HUNGARIAN HOSPITAL MODBIDITY STUDY AND SYSTEM SIS79/GENERA

In Hungary, representative hospital morbidity studies have been in progress (including each hospital and department) since 1972. Data of the inpatients are collected yearly with a sampling rate from 10 to 50 percent. It amounts to information on 200 to 600 thousand patients per year.

Processing of a rather large sample (600 thousand records, 60 to 80 million bytes) was to be accomplished on a comparatively small machine. The requirements were rather complicated and subject to modifications from time to time. At first, the machine used was CDC-3300 having 64 K words of memory with two 8 Mbyte disc units and two or three tape units available. The machine was overloaded so we could run small jobs (some minutes of CPU time) only. Consequently, jobs utilizing the total sample were to be run rarely. It became necessary to examine questions concerning the strategy of data processing. On the other hand, the data set was to be divided and compressed. The running time of the job (as well as other resources: memory, disc, tape) was to be minimized.

Later we got access to higher-capacity machines [15] (a HwB 66/60 or two HwB 66/20 with 100 Mbyte discs and 256 K words of memory). The problem of capacity became less important. But taking into account the requirements of conversational processing and the aspiration to faster turn-around in bacth processing (and the expenses as well) optimization of storage and time were expedient.

Let us outline the basic methods utilized to achieve shorter run-time in statistical investigations. First, we created statistical tables from frequencies and cumulated values instead of the original data. The tables were obtained in some seconds, practically independently of the sample size [15]. Let us note here that the hospital morbidity study required descriptive statistics mainly: tables contained frequencies, cumulated values and some simple rates (e.g. morbidity rate, etc.) and basic characteristics of the distribution (such as mean, standard deviation, range). More complicated statistical analyses do not make rise to new situations concerning fast processing of a large amount of data. Usually, mathematical statistics need frequencies and cumulated values (sums, quadratic sums, sums of products) (see statistical literature [11]). Then these values can be processed e.g. by SPSS programs.

Another method applied was a general technique in programming. The programs of the system are generated in each case depending on the parameters of the task. This technique (based on earlier experiences) was consistently applied in system SIS77 developed on HwB 66's [21]. In this improved version (in SIS79/GENERA) this technique was developed further [13]. The task of generating was placed under control of a general purpose system (GENERA) improving integrity and efficiency of the system. System GENERA and the possibilities provided by statistical system SIS79 will be dealt with in later sections.

In the Hungarian Hospital Morbidity Study, statistical systems SIS77 and SIS79 provide quick access to data and detailed analysis even for individual researchers. Even in the case of large mass of data and complicated conditions the system needs modest resources only. A COMECON-project on juvenile hypertension coordinated by the National Cardiology Institute that was successfully accomplished by systems SIS77 and SIS79.

In statistical tasks (especially in large and complicated systems) the method of sequential processing is suitable. Sequential processing is similar to sequential sampling [23] known from mathematical statistics. It produces a more and more widening sphere of information depending on the information obtained before. But compared to sequential sampling it does not mean an increasing amount of information of the same kind; in this case the kind of information is subject to change as well. Users (doctors, economists, etc.) first receive simple, easy-to-survey data (tables, graphs, descriptive statistics). The more and more detailed questions are based on the information obtained earlier and can optionally be answered on the base of a widening population. (That is a wider subset of the data set.) In this way needless information is not to be gathered, simple relations are enlightened immediately and the user gets an overall picture of the sample investigated. This method provides means for obtaining more valuable information from the data available.

Determination of code values for data is another interesting and important problem in data processing. It may require mathematical statistical investigations as well as representative sampling. One of the problems in the hospital morbidity studies was producting a reliable identifier for patients. A comparatively short, easy-to-code identifier was required with negligible probability of accidental coincidence (incorrect identification). The task of selecting the representative sample was a problem of similar complexity. Sampling based on the birthday of inpatients proved to be quite uniform [3]. Usually, representative sampling from individuals of multiple occurences is a complex matter requiring complicated mathematical investigations [10].

Multiple hospitalized persons and inpatients having multiple diagnoses require a file organization different from usual statistical data bases. The elements to be examined arenot the original records (hospitalized cases). New basic elements (one multiple hospitalized person or one diagnosis) are to be constructed. Problems of this kind are directly connected to data bases (to relational data models [1,4] especially).

3. PROGRAM GENERATING

System GENERA is a system to build generator programs having subsystems. Subsystems can have a set of parameters, they are given value by unified and flexible methods. A generator system based on GENERA has a predefined host language (or a set of host languages such as FORTRAN or PL/1). Text to be processed consists of host language statements and GENERA directives. Former ones become statements of generated program without any modification. On the other hand, the appropriate text generated by the designated subsystem replaces the directive.

The example in *Fig. 1* illustrates a source file of a generator system. Function of directives is not be explained here, for details see the following sections of this paper and [12] describing the generator system

3.1. STRUCTURE OF A SYSTEM BUILT UP ON G E N E R A

A system based on GENERA integrates any number of generator procedures to make a precompiler. These procedures form subsystems of the generator program and are called into execution by entering a directive onto the source file. Detecting a directive control is passed to the main entry point of the subsystem to read in parameters. Then the subsystem is executed. Having completed its function, subsystems return control to the main program to continue processing of the source file.

OPTION is a subsystem of program control (see example on *Figure 1*). It can be executed as the first step of a GENERA run and initializes some global variables of system to achieve a non-standard handling of source lines. The user controls the structure and contents of output information (generated program and listings) by OPTION.

A preprocessor subsystem (PREP) is contained in batch oriented versions of GENERA. This is a subsystem that cannot be called in by the user directly, and is always executed prior to any other functions of GENERA. Each line of input is examined, lines containing directives or parameters are checked. Statistics of the recognized directives are collected, and the unrecognized ones are reported. Then the parameters are tested if they meet the rules defined for the subsystem. Having found an error, the run is terminated abnormally at the end of preprocessor phase. The preprocessor performs transformations on parameter descriptions to provide an interface between a user--oriented description scheme and the program requirements. It can make both the programming of subsystems and definition of parameters easier.

As GENERA processes a number of input files (primary input containing host language program, directives and parameters; secondary input file containing specially structured data for certain subsystems; job generator (JOBGEN) input file describing non-standard job-setup) an Initial File Conversion Subsystem (IFCS) accompanies the system. IFCS builds up the input files from a single input file (MIXEDIN) and it can include some additional features (such as selecting given disc files or tapes as parts of input file) depending on the possibiliites provided by the operating system.

4. STATISTICAL INFORMATION SYSTEM SIS79/GENERA

System SIS77 and SIS79 have been mentioned before. The most important procedures of SIS79 will be presented here.

4.1. DATA TRANSMISSION AND CONVERSION

In a system to handle a great mass of data, efficiency of input-output operations is important. We developed a pair of I/O statements (#LECTOR, #SCRIPTOR) to perform these operations They are given the record structure (name, length and type of each field), and a program-fragment is generated to read or write the annotated variables.

The example in *Figure 1* shows an input directive #LECTOR. The meaning of the set $PARAM goes without saying. The set $DESCR gives format for reading record named PATIENT (COBOL--style level numbers and FORTRAN format items are used).

Procedures to generate I/O operations are needed in some systems because high-level languages analyze format specifications in run-time. Formats are usually not changed while running the program, so run-time evaluation is not needed. However, compilers do not translate format items to machine code.

Our input procedure generates a set of host-language statements to read the record 'as-is' (without any conversion) and to select and convert values of variables using efficient character-handling routines. Hence, FORMAT items are evaluated in compile-time instead of run-time. A large amount of processor time can be saved if there are I/O statements frequently used in the program. The method is especially useful in FORTRAN programs.

4.2. COMPRESSED BINARY STORAGE

Data storage can be a problem of great importance in some statistical systems. Let us see the following example. A large amount of data is to be stored on mass storage devices. It is known that data set contains numbers of small values. These numbers can be described by one or two decimal digits but they are freqently used and character form requires a conversion to be performed each time the data are read or written. On the other hand, data stored in binary form can be read or written without any conversion but in this case each number requires a full word of storage. (It is right for word oriented machines only.) We should find a method that is efficient in both means. That is, it should provide a fast conversion and data should not occupy superflouos storage space.

The compressed binary representation used in our system reduces the storage space required while processor time used is not increased significantly. It is achieved by compressing length of binary form to the number of bits required to contain the greatest allowable value of variable. The compressed binary read and write procedures generate a program-fragment performing I/O operationand compression or decompression.

4.3. DATA TRANSFORMATION AND GRAPH REPRESENTATION OF FUNCTIONS

Data preparation tasks involving transformations (coding, analysis of functions) are included in this group. To perform these tasks we have to describe the transformation procedure itself. It does not cause any difficulties in the case of functional dependencies defined by simple formulas. On the other hand, code-tables can be extremely large, larger than the total amount of core memory available on the machine used. Description, control and storage of these tables can cause hardly resolvable problems. One of the installed generator systems based on GENERA, system SIS79, involves a certain storage method especially designed to be used in generated programs.

Using this method, functions or transformations defined by code-tables are described in the form of a hierarchical graph [13,15]. This graph is divided into levels corresponding to arguments of the function. A level contains one or more subtables controlling values of the variable belonging to given intervals. Being empty parts or identical segments included in the table, this method can provide a significant reduction of storage required for the table. Moreover, an efficient program can be generated to read and analyze the graph. While the necessary storage capacity is radically reduced (e.g. in a system used by the Health Service, tables based on the international code system of diseases were reduced to 5 percent of size, approximately), compute time did not increase essentially as compared to the time required for the method using a unique large table of values. Reduction of storage and run-time contains several interesting problems of graph theory and finite projective geometry [16].

Figure 1 contains two consequtive GRAPH directives. The first, "AGE CODING", codes variable AGE to variable CDAGE using one code table (SACKNO=1, LEVELS=1). Maximal value allowed for AGE is 100 (UPPBOU=100). Variable NUMBER is used for signaling errors. Second procedure "CONTROL" checks variables CDAGE, SEX, MAINCD and SUBCD using a graph of 4 levels and 34 elementary tables.

The tables are filled up by a general procedure contained in systems SIS77 and SIS79. Several advantages are obtained using this subroutine: the method applied to fill up the tables is the most compact and comfortable one, appropriate security is provided by the syntax analysis of table descriptions and detailed error messages. This subroutine provides means for a quick and easy calculation of some multivariate functions as well. We demonstrate the method to construct a graph on *Figures 2-4* using a very simple function. Table generating statements (on *Fig. 4*) contains (left to right) command codes, table identifiers or table values, index values and optional comments. Negative values are pointers.

4.4. EVALUATION OF LOGICAL EXPRESSIONS

Performing a statistical analysis, sometimes, data base should be divided into parts meeting requirements of the subsystem to be used. Decision rules dividing the data base are usually described by logical expressions of high complexity. In system SIS79 a generator procedure is applied to provide a simple method for defining these rules and to generate a program fragment performing the selection. (On mathematical logical investigations concerning this topic a lecture was given by I. Ratkó in Salgótarján, Hungary at the Conference on Mathematical Logics in Theory of Programming.)

We performed interesting investigations in probability theory concerning the problem of selection [13] to find optimal strategies of file dividing.

4.5. TABLE-FILES, OUTPUT TABLES

Results obtained by statistical data processing do not contain the data of individual items but those of typifying ones. Thus the files consisting of these raw data must be transformed into that of statistical data (frequency characteristics, code values totals, quadratic sums, product sums, etc.). Consequently, in statistical information systems it is not advisable to apply the languages developed particularly for handling and querying processes of raw data items. We achieved that after a suitable preprocessing (creating 'table-files') a lot of different output tables can be obtained using a few seconds of CPU time (on HwB 66/60) independently of the size of the sample. It makes possible to perform statistical study of large sample in interactive mode, too.

5. REFERENCES

[1] E.G. Codd, "A Relational Model of Data for Large Shared Data Banks", Comm. ACM, Vol. 13., 1970, pp. 377-387.

[2] E.G. Coffman, P.J. Denning. Operating Systems Theory, Prentice-Hall, 1973.

[3] M. Csukás, L. Greff, A. Krámli, and M. Ruda, "An Approach to the Hospital Morbidity Data System Development in Hungary", Colloques IRIA, Tome 1, Informatique Medicale, 1975, pp. 381-390. (paper presented at the Symposium on Medical Data Processing, Tolouse, 1975)

[4] J. Demetrovics, "On the Equivalence of Candidate Keys with Sperner Systems", Acta Cybernetica, Vol. 4, No. 3, 1979, pp. 247-252.

[5] J. Demetrovics, E. Knuth and P. Radó, "Specification Meta Systems", Computer, May 1982, pp. 29-35.

[6] D.E. Denning, P.J. Denning, and M.D. Schwartz, "The Tracker: A Threat to Statistical Database Security", ACM Transactions on Database Systems, Vol. 4, No. 1. 1979, pp. 76-96.

[7] W.J. Dixon, M.B. Brown (editors), BMDP Biomedical Computer Programs (P-series), Uniersity of California Press, Berkley, Los Angeles, London, 1979.

[8] M. Finkelstein, "A Compiler Optimization Technique", Computer Journal, Vol. 11, No. 1, 1968, pp. 22-25.

[9] J. Foisseau, R. Jacquart, M. Lemaitre, M. Lemoine, J.C. Vignat, and G. Zanon, "Program Development With or Without Coding", Software World, Vol. 12, No. 1. 1981, pp. 9-12.

[10] L. Greff, A. Krámli and J. Soltész, "The Modeling of the Sampling Procedure for the Hungarian Hospital Morbidity Studies", Modeling Health Care Systems (ed. E. Shingan, P. Aspden, P. Kitsul), IIASA, Laxenburg, Austria, 1979, pp. 172-177.

[11] M.G. Kendall, A. Stuart, The Advanced Theory of Statistics Vol. I-III, Griffing, London, 1958, 1961, 1966.

[12] P. Kerékfy, "GENERA - A Program Generator System", Progress in Cybernetics and Systems Research, Vol. 11., Hemisphere, Washington, 1980. (paper presented at the Fifth European Meeting on Cybernetics and Systems Research (EMCSR'80), Vienna, 1980.)

[13] P. Kerékfy, A. Krámli, and M. Ruda, "SIS79/GENERA Statistical Information System", Progress in Cybernetics and Systems Research, Vol. 11., Hemisphere, Washington, 1980. (paper presented at the Fifth European Meeting on Cybernetics, and Systems Research (EMCSR'80), Vienna, 1980.)

[14] E. Knuth, P. Radó, and A. Tóth, "Preliminary Description of SDLA", Tanulmányok - MTA Számitástechnikai es Automatizálási Kutató Intézete,105/1980.

[15] A. Krámli, M. Ruda, M. Csukás, and M. Galambos, "Large Sample Size Statistical Information System for HwB", Data Analysis and Informatics, ed. E. Diday, North-Holland, 1980, pp. 457-462. (paper presented at the Second International Symposium on Data Analysis and Informatics, Versailles, 1979.)

[16] A. Krámli, P. Lukács, and M. Ruda, "Probabilistic Approach to the Performance Evaluation of Computer Systems", Proceedings of the Third Hungarian Computer Science Conference, Vol. I, Invited papers, Budapest, 1981. pp. 51-64.

[17] N.H. Nie et al., SPSS Statistical Package for the Social Sciences (end edition), Mc Graw-Hill, 1975.

[18] J. Nievergell, "On the Automatic Simplification of Computer Programs", Comm. ACM, Vol. 8, No. 6, 1965, pp. 366-370.

[19] B. Perron (ed.) et al., IDMS Concepts and Facilities, Cullinane Corporation, 1977.

[20] M. Ruda, "Some Estimates in Connection with the Critical Path Method", Project Planning by Network Analysis, Proceedings of the Second International Congress (ed. H.J.M. Lombaers), North-Holland, Amsterdam, 1969, pp. 207-215.

[21] M. Ruda, "Statistical Information System with Health Service Application", MTA SZTAKI Tanulmányok, 87/1978, pp. 167-172. (paper presented at the Fourth Winterschool of Visegrád on the Theory of Operating System, Szentendre, Hungary, 1978.)

[22] M.D. Schwartz, D.E. Denning, P.J. Denning, "Linear Queries in Statistical Databases", ACM Transactions on Database SYstems, Vol. 4, No. 2, 1979, pp. 156-167.

[23] A. Wald, Sequential Analysis, Wiley, New York, 1947.

ÖSSZEFOGLALÁS

MEGJEGYZÉSEK A STATISZTIKAI ADATFELDOLGOZÁSSAL KAPCSOLATBAN

J. Demetrovics, P. Kerékfy, A. Krámli, M. Ruda

A dolgozat a statisztikai adatfeldolgozásban használt program-generáló eljárásokat ismerteti.

ОБ ОБРАБОТКЕ СТАТИСТИЧЕСКИХ ДАННЫХ

Я. Деметрович, П. Керекфи, А Краммли, М. Руда

Изучается система генерирующая программы для обработки статистических данных.

Közlemények 30/1984 53-59

DATA ENTRY, A VERY IMPORTANT PROCESS

ELENA BRAGADO, MIGUEL FONFRIA, EUGENIA MUNIZ

ICID, Havana, Cuba

Data entry process is one of the most important steps in data processing systems and the reliability of the results obtained depends upon the quality of the data entry process. Although computer techniques have changed rapidly in the last few years, many systems designers are still thinking in terms of punched cards. But input designs must be up to the level of current hard- and software.

Conventional data entry process involves information keying, the verification through repetitious keying, and the input to the system through a program which checks the validity of that information. From this validity checking we get a subset of the information which is declared wrong and it must be keyed once more. This process goes on again and again, until the checking program finds no errors.

This process is generally present in most of the data processing systems and it is easy to realize the great length of time spent on it, as well as the amount of non-reutilizable support wasted on data entry by punched cards.

Substitution of electromechanical equipment for data entry is thus necessary because of, among others, the following reasons:

- increase of the amount of data to be processed
- increased price of the data support, which is not reutilizable
- the necessity of increase speed

the necessity to guarantee the quality of primary data, and not only in order to facilitate error detection but also to avoid generation of new errors.

The development of computer techniques itself has determined that data entry on magnetic surfaces becomes a substitute for key-punched data entry.

At present, those systems, have many features and consequently they have extraordinarily increased their potentiality.

Magnetic surfaces for data entry have the following advantages:

- they are re-utilizable
- they become cheaper with each year that passes and have an increased capacity for information
- they can be updated, thus allowing checks - with detection and correction of errors - at the time of data input.

The actual fact at the present time is that in data entry process punched data is not used at all or is almost unused in many countries.

At present, there are different equipments and systems for data entry, depending on the different requirements of the users, and they provide different levels of data entry manipulation.

Data entry in Cuba

In Cuba we have the same problem with the same characteristics. Also, we have been trying to perform the data entry process in the best way. In order to accomplish this, there are two strategies:

a) off-line data entry

b) on-line data entry.

a) Off-line data entry won't be discussed here because it does not constitute a subject in this issue.

b) On-line data entry

Under this strategy the Multi-Terminal Data Entry System (COPDAT) was developed. This is a specific operating system oriented to data entry and its validation on Cuban minicomputer analog to PDP11/20.

This system allows working with up to 16 terminals connected via multiplexor, it being possible to create from 1 up to 16 different files at the same time. This means that each terminal may create on file or several terminals may be associated in order to get information to create the same file.

The temporary or final result may be stored in magnetic tapes, in OS or DOS format. It is possible to verify a file totally or partially and also to verify it from the beginning or from a given record. Besides, it is possible to validate a file from any other input equipment of the configuration. All of these functions may be performed simultaneously.

File creation may be controlled by commands which immediately validate the input information, and also files can be created in several working-days. During interactive input, errors can be detected and also they can be corrected immediately.

When each terminal finishes its labour, some operator statistics are shown in the system console. They are: terminal number, total keystrokes, total records, total errors, beginning time and ending time.

A listing is also supplied with wrong records, each with the terminal number on which the record was typed. In this listing, wrong fields are signalized with asterisks under the wrong characters. These listings are obtained through a spooler, and each listing is identified with the name of the corresponding task.

COPDAT guarantees, in case of system failure, all the information typed up to the moment of the failure.

COPDAT supplies also other auxiliary functions, very useful in the development of the work. These functions are: listing of disk's directories, equipment initialing, truncation of a task, and file deletion.

COPDAT enables checking if a field is numeric, alphabetic, alpha-numeric or symbolic; if it is equal or different from given characters of fixed values. COPDAT can also check that a given string does not appear as a substring of a field. Range checks, arithmetic relationships, sum checks and inter-field dependencies can all be specified. As a result, verification on central computer can be further reduced or even eliminated.

COPDAT implementation

COPDAT is divided into the following main modules:

- task supervisor
- multi-terminal handler
- memory allocator
- command executer
- file control system.

Task supervisor decides which tasks are going to be executed and when. In order to take this decision it uses the roundrobin method with priorities. These priorities are: the highest one for interactive tasks, the second one for non-interactive tasks, and the lowest one for auxiliary functions.

Multi-terminal handler inquires into the terminals and achieves all treatments about them.

Memory allocator is one of the fundamental modules of COPDAT because it distributes the available memory for the execution of different tasks. Memory allocator uses the FIRST FIT method, and also it tries to group the released memory into greater available space blocks in order to avoid fragmentation. When the available blocks do not satisfy the memory request, this memory allocator checks whether the sum of all blocks together satisfies the request and then it performs a memory condensation.

Command executer does the validity checks, and the *file control system* treats all peripheral equipments in the configuration.

CONCLUSIONS

In order to obtain more efficient data processing systems, it is very important to pay the necessary attention to data entry process.

COPDAT is the implementation of an on-line data entry system for Cuban minicomputers.

Using this kind of systems the data required to run your business is processed sooner - and in data processing it is very useful to save time.

BIBLIOGRAPHY

[1] Data IV Intelligent Data Entry.
Four-Phase Systems.
Cupertino, California. 95014.

[2] Gilb, Tom, and Gerald Weinberg. Humanizing Data Entry by Default.
Datamation, Aug. 1976, 73-76.

[3] Knuth, D. The Art of Computer Programming
Vol. 1.

[4] IBM Corporation. "3740 Data Entry System"
Auerbach 302. 4239.100.

[5] Key to Stroke. Auerbach 302.0000.200.

[6] Demetrovics, J., Gyepesi, Gy. On the functional dependency and some generalizations of it.
Acta Cybernetica, 5(1981)3, 295-305.

[7] Demetrovics, J., Knuth E., Radó, P. Specification meta systems . Computer, May (1982) 29-35.

ÖSSZEFOGLALÁS

ADAT-BEVITEL, EGY NAGYON FONTOS FOLYAMAT

E. Bragado, M. Fonfria, E. Munis

A cikk indokolja az adat-bevitel fontosságát a számitástechnikai folyamaton belül; áttekinti a Kubában használatos adat-beviteli technikákat; és ismerteti a Kubában kifejlesztett /PDP11/20-al analóg/ mikroszámitógépre megirt adatbeviteli rendszert /COPDAT/.

ВХОД ДАННЫХ, ОЧЕНЬ ВАЖНЫЙ ПРОЦЕСС

Е. Брагадо, М. Фонфриа, Е. Мунис
Хаванная, Куба

В статье кратко описывается важность процесса входа данных в вычислительной технике и разные методы,использованные на Кубе. Мы познакомимся с системой входа данных /COPDAT/, разработанной для кубинского Микро-помпьютера типа PDP.

Közlemenyek 30/1984 61-71

SOME EXPERIENCES IN THE DEVELOPMENT OF DBMS [1]

P. DIPOTET

Institute of Mathematics, Cybernetics
and Computing Sciences
Havana, Cuba

Both, United Nations REPORT-71 and the first Intergovernmental Conference on Strategies and Policies (SPIN) organized by UNESCO and IBI in Spain in 1978, underlined the importance of developing a strategy for informatics in order to make the best use of the domestics resources of the country.

In Cuba, a national organization, "The Institute for Management Information Systems and Computing Techniques", has been created to coordinate cuban efforts in research, development and applications of computer science (INSAC).

Some important points in Cuban Informatics strategy are:

- the high priority given to the application of informatics in the development and control of the national economic plan;
- the development of the national statistical information system;
- the development of some special mini and microprocessor systems and software packages applied to some services and mainly to health services;
- the R/D works in software and hardware and the education and training of computer specialists supporting the former three application lines.

Next we present some of the most important experiences and approaches to the development of DBMS in our country.

---

[1] Received on April, 1982

APPLICATIONS OF DBMS TO PLAN PROCESSING

In the Electronic Processing Division of JUCEPLAN (Cuban Central Planning Board) a data model together with its retrieval language has been developed in order to be used by informatics experts working in the United Systems for Plan Processing (SUPP).

SUPP objectives are*:

- To reduce the global data processing time for daily tasks.
- To normalize the supports and media for developing, implementing and processing the functional subsystems.
- To unify the Computer Center's data base, with the possibility of being up-dated and consulted through diaplays.
- To achieve data integrity and protection in all data processing phases and stages.
- To improve the working efficiency of the whole computer center's staff.

Good ideas on relational data models have been used in order to derive the data model to be used by computer specialists in the Computer Center to implement the new processing demands for the information contained in the data base.

In SUPP a sharp differentiation between retrieval and up-dating language is done, in order to reduce the requirements stated to each one of them.

The former fact and the possibility of non-instantaneous retrieval allow us to simplify the computer implementation of the data model and their corresponding languages.

The set of relations present in SUPP may be divided in the following way:

- The classifiers, formed by the list of codes of the objects to be identified by the system, their descriptions and other informations associated to them.

---

* Notes are taken from: BARRERA J. et. al. "Use of the Relational Model within the Data Base of Plan Processing System". Economia Planificada III/2/1980, La Habana.

The classifiers are relations with simple primary keys (only one domain)

- The planning indicators, formed by the ecomonical data needed for the elaboration of the plan. They are identified by the code sets defined in the classifiers.
- The attributes of the planning indicators. They are formed by other informations needed by the exhaustive definition of the indicators. They are identified by the codes defined in the classifiers.
- A combination of the former relations.

The first three subsets are the updatable relations.

These subsets allow any information to be stored or retrieved in SUPP, to be in the third normal form defined by Codd.

OPERATIONS IN THE RETRIEVAL LANGUAGE

The retrieval language is an algebraic one. The proposed operations are a selection of the ones described by Codd. There is also the operation "transposition". This operation set allow us to derive any meaningful relation using only the relations stored in SUPP.

The operations with only one relation are: projection, restriction and transposition. We will define the last one. The others are similar to Codd's ones.

The transposition operation allows us to change the representation of the relation. There are two variants:

- to disgregate in various M-tuples a group of indicators defined in the same N-tuple;
- to fuse in one N-tuple various indicators defined in one group of M-tuples.

The operations with two relations are: union, intersection and difference which are very similar to Codd's ones.

PERSPECTIVES

A group of specialists of this organization (JUCEPLAN), is developing (under the scientific direction of Academic Lavrov) a DBMS using some ideas of Codd's relational model. There are some interesting results in this work which is expected to be finished in the second half of 1983.

Another group is working in the improvement of SUPP present version. They are dealing with some semantic problems related to the stored information. They are also trying to implement some concurrent processing facilities and the bank of method concepts.

A complete information about these works is possible to read in: "Using the Relational Model within the Data Base of Plan Processing System", Rev. Economia Planificada III/2.

APPLICATION OF DBMS TO CUBAN STATISTICAL OFFICE

The aim of the work (1) is to set Cuban requirements for statistical data bank considering GDR experiencies and our conditions and planned development of statistical service in Cuba.

The first thing to be considered is that the data bank should be constructed under the frame of the Automated System for State Statistics of Cuba, a project being designed now. The second aspect is the five year strategy plan for the development of statistical service that was recently approved in the Cuban State Committee for Statistics.

The work brings some results in the field of data bank, information modelling, general architecture, and a methodology for the formal description of the informative model.

The proposed topics being considered in this work are:

1. Definition of Cuban requirements for the statistical data bank from the point of view of the information needed (indicators, micro- and macro-data, registers etc.) and form the point of view of user requirements

for output results, evaluation and analysis of data etc.. At this point the statistical branch to be considered in the data bank (industry, buildings etc.) has to be decided.
To carry out this work, the experience of GDR data bank is considered, specially in the field of data bank statistical service.

2. The actual state and future trends in the development of data banks. After having the general requirements for Cuban data bank, a review of the application of data banks, - mainly in CMEA countries -, is done in order to gather experiences and to point out the main trends in this field.

3. A definition of a databank management system as a high integrated system for storing, retrieval, evaluation and analysis of statistical data should be given.

   See Hernández, O.; Lastra, O.; "Definition of Requirements for the Cuban Statistical Data Bank System" - Cuban Statistical Office. March. 1981.
   In this sense the main idea was pointed out in paper "A brief contribution to the study of statistical information, systems architecture" presented in ISIS'78 seminar containing basic theoretical principles of what has been recently developed as interface between data bases and bank of methods.
   An important result of this is the definition of the informative model as the formal description of concepts, semantic of data, user requirements for evaluation and analysis of data etc.. This theoretical part would not be the main part of the work but it is considered important to give support to the rest of the work.

4. Development of a methodology for the description of the informative model.

5. Application of the methodology to describe the informative model for the Cuban statistical data bank, according to the requirements and goals defined in the first point. After this being done we will have the complete view of the data bank to be implemented in Cuba from the information point of view in terms that could be comprehensible for statistical users and could serve as discussion tool and as the basis for the construction of the data bank.
6. Description of the software requirements and functions of a Data-Base System for Cuban Statistic (DBSCS).
7. Advantages of SPAZ as data base management system of the DBSCS. The capability of SPAZ to store data, according to the types and requirements of the information, is quite explicit, as well as other factors, like integrity, protection etc..

The system provides such advantages for the evaluation and analysis of data stored in the data bank, as:

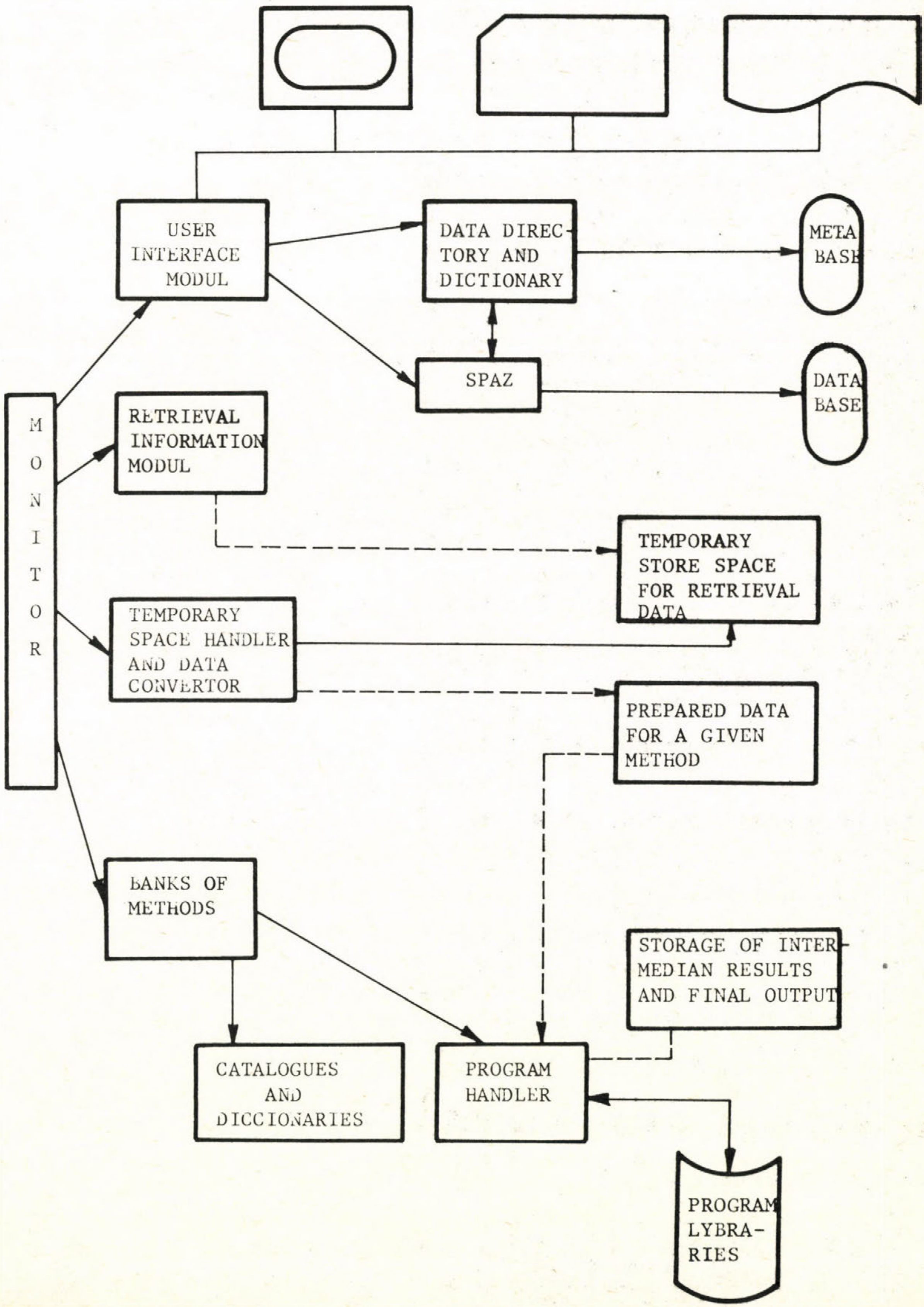
- mathematical computation and matricial algebra;
- descriptive statistics;
- index number;
- preparation and printing of complex output tables;
- graphics;
- regression and correlation analysis;
- time series analysis and forecasting;
- econometrics;

and others that can be included.

In fact, further versions of MGCE will approach a bank of methods specially in the fields of regression, time series analysis forecasting, and econometrics.

This approach will also guarantee the work to give practical results, since we have version 1.3 of MGCE working in user interface moduls and the retrieval programs would be themes for other works.

Definition of DBSCS, it general architecture and components.



DBMS IN THE CUBAN ACADEMY OF SCIENCES

In the Institute of Mathematics and Cybernetics of the Cuban Academy of Sciences, the works in DBMS are oriented in two main directions: DBMS for minicomputer and the development and/or implementation of well known systems as, for example, the Hungarian SDLA[1].

- *DBMS for minicomputers*

  This work is done in cooperation with the Computer Center of the USSR Academy of Sciences. The system being developed is called SINOD.

  The SINOD has twoo main blocks, an adaptive dialogue system (ADS) and a data base (BD), both are written in FORTRAN for the CM-4.

  The ADS has three main modules: SYNTAX, MONITOR and INTERP. The firstone transforms the input text into a normalized format. The MONITOR analyzes the condition appearing on each line of the program written in the transformation language of the ADS and when the condition is valied then it passes the control to the INTERP.

  The INTERP performs the actions corresponding to the conditions analyzed by the MONITOR. The arithmetic operators used in the transformation language are: +, -, *, /, with the same meaning they have in the common programming language.

  The transformation language has also the following transfer operators:

  PUT; for transferring a read data to one memory address called INPUT,

---

[1] Knuth, E., Preliminary description of SDLA. Tanulmányok, 105/1980. MTA SZTAKI, Budapest

STOR; to transfer one arithmetic operator from the memory address INPUT to the memory address OP,

and the following operators executing some actions:

EXEC; doing effective the operation indicated by the arithmetic operator,

EXIT; ending the program execution.

SINOD is being applied to the development of a data base system for planning the sugar cane harvest in a socio-economic region.

We expect the system to be finished in the middle of 1983.

- *Another development of DBMS*

With the support and cooperation of the SZTAKI it has being installed in the Computer Center of the Cuban Academy of Sciences, the Hungarian version of the ADBMS, a special CODASYL type data base management system originally developed at the University of Michigan and improved at SZTAKI.

This system consists of 175 FORTRAN and 29 assembly subroutines. FORTRAN and COBOL can be used as host language of ADBMS commands. The commands can be invokated from a used program by CALL. The schema possibilities are limited compared with those of CODASYL approach.

There is no possibility to define a subschema and also to access concurrently the same data base at the same time by different users, but the ADBMS contains the main features of CODASYL and also some extensions to it. The ADBMS makes possible it to get practice in data base management systems and also it can be applied for problems having not too large mass of data (e.g. 15000-20000 records).

Formerly the strategy of locating records in ADBMS was the following: records are put into the pages of data base essentially in the order of arrival, sequentially. Modifications made in the Computing and Automation Institute in Hungary completely altered this strategy. The solution was a hash algorithm by which the physical address of record is computed. So looking for a record does not mean several pages replacements between the storage device and the main memory, because using the computed physical address, the systems can find the page required only with one page replacement. In such a way the number of page replacements during the search of a record was decreased, and the system became more efficient.

Previously the load of a mass of records under certain circumstances required a total time of 45 minutes on an EC-1020 machine. After the modifications, under the same circunstances, the total time of the same load was 9 minutes.

DBMS IN THE INSTITUTE FOR THE DEVELOPMENT OF MANAGEMENT INFORMATION SYSTEMS AND COMPUTER TECHNIQUES (INSAC)

The main works in DBMS are related to the implementation of the SOMIS which is a DBMS with several important limitations. Direct access in SOMIS is provided by chained lists which may be structured in a hierarchical way. Users can take advantages of this hierarchies in order to avoid redundant records.

SOMIS uses direct access files which separate the data in two classes. Master files contain one of the classes and are ordered in logical sequences according to the keys. The second group of files, linked files, contain the other class of data. These files are organized as chained lists with variable record lengths. Each list is associated to a record in the master file. It is possible to access the Base using COBOL, PL/1 and ASSEMBLER.

SOMIS main difficulties are the low speed and the high frequency of maintenance. These maintenances are needed because of thw two related files and the fact that changes in one file are reflected in the other one.

The low speed is a consequence of the linked lists. The first record is accessed in a direct way (via its key) and the others are accesses sequentially. Obviously the search by multiple keys on big files may be very slow.

For this reason SOMIS is not recommended for on-line application with big files and critical response times.
There are other important results in DBMS which are going to be published in near future.

ÖSSZEFOGLALÁS

ADATBÁZISKEZELŐ RENDSZEREK A KUBAI NÉPGAZDASÁGBAN

Perfecto Dipotet

A szerző ismerteti a Kubában használt információs rendszerek közül a legjelentősebbeket. Felhasználói szemszögből elemzi őket, de rövid funkcionális leirást ad szerkezetükről is.

ОПЫТ ПРИМЕНЕНИЯ СУБД В ЭКОНОМИКЕ КУБЫ

Перфекто Дипотет

Дается обзор информационных систем разработанных и применяемых на Кубе. Системы анализируются с точки зрения пользователя, но автор пытается дать и краткое описание функциональных характеристик.

Közlemenyek 30/1984 73-87

REMARKS ON CLOSURE OPERATIONS

VU DIC THI

Computer and Automation Institute
Hungarian Academy of Sciences

§.1. INTRODUCTION

The relational datamodel was defined by E.F.Codd [3]. Many papers have appeared since that dealing with the combinational characterization problems of functional dependencies.

The main purpose of this paper is to investigate the connection of closure operations with the minimal keys and antikeys.

§.2. DEFINITIONS

In this section, we present some necessary definitions.

Definition 2.1. Let $X = \{1,\dots,n\}$. The function $F: 2^x \to 2^x$ is called a closure operation if for every $A,B \subseteq X$

(i) $A \subseteq F(A)$ (extensive)
(ii) $A \subseteq B \Rightarrow F(A) \subseteq F(B)$ (monotone)
(iii) $F(F(A)) = F(A)$ (idempotent)

Let M be an $m x n$ matrix and X be the set of its columns. Let $F_M(A)$, $A \subseteq X$, be a function such that $F_M(A)$ contains the ith column of M iff any two rows identical in columns belonging to A are also equal in the ith column.

It is clear that $F_M(A)$ is a closure operation.

Definition 2.2. Let F be a closure operation. We say that M represents the closure operation F if $F = F_M$.
It is known [1] that any closure operation is representable by an appropriate matrix M.

Definition 2.3. Let F be a closure operation and $A \subseteq X$. A is a key of F if $F(A) = X$.

Definition 2.4. Let F be a closure operation. We define

$$K_F = \{A : F(A)=X,\ (\forall B \subseteq A)(F(B)=F(A) \Rightarrow B=A)\}$$

That is: K_F is a set of minimal keys. We say that an $m \times n$ matrix M represents the family K iff $K = K_{F_M}$.
It is easy to see that the family of keys of a closure operation create a Spernér-system.
We denote $\Delta(K) = \min\{m : K = K_{F_M} : M \text{ is an } m \times n \text{ matrix}\}$.
where K is a Sperner-system over X.

§.3. THE PROPERTIES OF THE CLOSURE OPERATIONS

It is easy to prove that if F is a closure operation and $A_i \subseteq X$ $(1 \leq i \leq m)$, then $F(\bigcup_1^m A_i) = F(\bigcup_1^m F(A_i))$ and $F(\bigcap_1^m F(A_i)) = \bigcap_1^m F(A_i)$.

Definition 3.1. Let F be a closure operation. We say that $A (A \subseteq X)$ is a maximal element of F iff for all $B (B \subseteq A) : F(B) = F(A)$ implies $B = A$.

Denote by $M(F)$ the set of the maximal elements of F i.e.

$$M(F) = \{A:(\forall B \subseteq A)(F(B)=F(A) \Rightarrow B = A)\}$$

Theorem 3.2. Let F be a closure operation. Then

$$M(F) = \{A:(\forall C)(A \subseteq F(C) \Rightarrow C \not\subset A)\}$$

Proof. Assume the $A \in M(F)$, but $\exists C$ such that $A \subseteq F(C)$ and $C \subset A$. We have $F(A) \subseteq F(F(C)) = F(C)$ by (ii) and (iii). $C \subset A$ implies $F(C) \subseteq F(A)$, so $F(A) = F(C)$ holds. Consequently there exists $C \subset A$ such that $F(A) = F(C)$. This conteadicts to the assumption $A \in M(F)$.

Now, assume that

$$\forall C \; (A \subseteq F(C) \Rightarrow C/A) \quad (*)$$

but $A \notin M(F)$. This means that there is a set B such that $B \subset A$ and $F(B) = F(A)$. (i) implies $A \subseteq F(A) = F(B)$. Consequently, there is B such that $A \subseteq F(B)$ and $B \subset A$. This contradicts the fact that A satisfies $(*)$. The theorem is proved.

Let $M_1(F)= \{A:A \quad F(A) \text{ and } (\forall B \subseteq A)(F(B)=F(A) \Rightarrow A=B)\}$

Denoting by M_n the extremum of $/M_1(F)/$ it can be proved that $\lim\limits_{n\to\infty} \frac{M_n}{2^n} = 1$, see [2].

We denote by N_n the extremum of $/M(F)/$. It is clear that $M_n \le N_m \le 2^n$, hence $\lim\limits_{n\to\infty} \frac{N_n}{2^n} = 1$.

Definition 3.3. Let F be a closure operation over X, we call the image $F(A)$ of A as a nontrivial one if $A \subset F(A)$.

Let $P(F) = \{F(A):A \subset F(A)\}$ and denote by P_n the extremum of $/P(F)/$.

Theorem 3.4. $P_n = 2^{n-1}$.

Proof. Let $T(F) = \{A : A \subset F(A)\}$. It is clear that $/T(F)/ \geq /P(F)/ (*)$. On the other hand (iii) implies $F(F(A))=F(A)$, so $P(F) \cap T(F)=\emptyset$ holds. Consequently, $/P(F)/+/T(F)/\leq 2^n$, we obtain $2/P(F)/\leq 2^n$ by $(*)$. Hence $/P(F)/\leq 2^{n-1}$ Take $b \in X$ and let $F(A)=A \cup \{b\}$ for every $A \subseteq X$. It is easily seen that F is a closure operation and $/P(F)/=2^{n-1}$. The theorem is proved.

§.4. THE CONNECTION BETWEEN THE MINIMAL KEYS AND ANTIKEYS

Let K be a Sperner-system. We define the set of the antikeys of K, denoted by K^{-1}, as follows:

$$K^{-1} = \{A \subseteq X : (B \in K \Rightarrow B \not\subseteq A) \text{ and } (A \quad C) \Rightarrow (\exists B \in K)(B \subseteq C)\}.$$

That is: the antikeys of K are the subsets of X not containing the elements of K and which are maximal for this property. Ot is clear that K^{-1} is a Sperner-system.

Remark 4.1. In [1,4], it has been proved that if K is an arbitrary Sperner-system then there exists a closure operation $F(F')$ for which $K = K_F$ $(K=K_F^{-1})$.

The antikeys play important roles for the evaluation of $\Delta(K)$ as well as for the construction of a concrete matrix representing a family K or for finding minimal keys.

The algorithm for finding the set of antikeys:

Let $K = \{B_1,\dots,B_m\}$ be the Sperner-system over X we have to contruct K^{-1}. For every $q = 1,\dots,n$, we construct $K_q = \{B_1,\dots,B_q\}^{-1}$ by induction.

Step 1: Construct K_1 in the following way:

$$K_1 = \{B_1\}^{-1} = \{X\setminus \{C\}:\ C \in B_1\}$$

Step $q+1$: Construct K_{q+1} in the following way:
By the inductive hypothesis we have constructed $K_q=\{B_1,\dots,B_q\}^{-1}$. Suppose that $X_1,\dots,X_p$ are the elements containing B_{q+1} of K_q. So

$$K_q = \{X_1,\dots,X_p\} \cup \{A\in K_q : B_{q+1} \not\subseteq A\}.$$

Denote $\{A\in K_q (B_{q+1}\not\subseteq A\}$ by F_q. For all i $(i=1,\dots,p)$ we construct the antikeys of $\{B_{q+1}\}$ on X_i in the analogous way of as in step 1, which are the maximal subsets of X_i not containing B_{q+1}. Denote them by $A^i_1,\dots,A^i_{R_i}$ $(i=1,\dots,p)$.

Let $K_{q+1} = F_q \cup \{A^i_T : A^i_T \not\subset A,\ if\ A\in F_q,\ 1\le T\le R_i,\ 1\le i\le p\}$

Theorem 4.2. $K_m = K^{-1}$

Proof. We prove the theorem by induction. The fact $K_1 = \{B_1\}^{-1}$ is obvious. Now we have to prove $K_{q+1} = \{B_1,\dots,B_{q+1}\}^{-1}$ using the induktive hypothesis $K_q = \{B_1,\dots,B_q\}^{-1}$. We have to prove:

a) If $A\in K_{q+1}$ then A is the subset of X not containing B_T $(T=1,\dots,q+1)$ and being maximal for this property.

b) Every $A \subseteq X$ not containing elements B_T $(T=1,\dots,q+1)$ and being maximal for this property is a element of K_{q+1}. The proof for (a): Let $A\in K_{q+1}$. If $A\in F_q$ then A doesn't contain any one in $B_1,\dots,B_q$ and A is maximal for this property and at the same time $B_{q+1}\not\subseteq A$. Consequently, A is a maximal subset of X not containing B_T $(T=1,\dots,q+1)$.

Let $A\in K_{q+1}\setminus F_q$. It is clear that there is A^i_T $(1\le i\le p$ and $1\le T\le R_i)$ such that $A = A^i_T$. Our construction shows that $B_l \not\subseteq A^i_T$ $(l=1,\dots,q+1)$. Because A^i_T is an antikey of $\{B_{q+1}\}$ for X_i, then $A^i_T = X_i\setminus\{b\}$ for some $b\in B_{q+1}$. Now it is obvious that $A^i_T \cup \{b\} \supseteq B_{q+1}$. If $a\in X\setminus X_i$ then, by inductive hypothesis, for

$A_T^i \cup \{a\}\ \{b\} = X_i \cup \{a\}$ there is $B_l\ (l=1,\ldots,q)$ such that $B_l \subseteq A_T^i \cup \{a\} \cup \{b\}$. X_i doesn't contain $B_1,\ldots,B_q$ by $X_i \in K_q$. Hence $a \in B_l$. If $(B_l \setminus a) \subseteq A_T^i$ then $A_T^i \cup \{a\} \supseteq B_l$. For every $B_l\ (1 \leq l \leq q)$ such that $B_l \subseteq X_i \cup \{a\}$ and $B_l \not\subseteq A_T^i$ we have $b \in B_l$.
Hence $(B_l \setminus \{a,b\}) \subseteq A_T^i$. Consequently, there is $A_1 \in F_q$ such that $A_T^i\ A_1$. This contradicts $A \in K_{q+1} \setminus F_q$. So there exists $B_l\ (1 \leq l \leq q)$ such that $A_T^i\ \ \{a\} \supseteq B_l$.
The proof for (b): Suppose that A is the maximal subset of X not containing $B_T\ (1 \leq T \leq q+1)$. By inductive hypothesis, there is $Y \in K_q$ such that $A \subseteq Y$.
The first case: If $B_{q+1} \not\subseteq Y$ then Y doesn't contain $B_1,\ldots,B_{q+1}$. Because A is the maximal subset of X not containing $B_T\ (1 \leq T \leq q+1)$, then $A=Y$. $B_{q+1} \subseteq Y$ implies $A \in F_q$. Hence $A \in K_{q+1}$.
The second case: If $B_{q+1} \subseteq Y$ then $Y = X_i$ for some i in $\{1,\ldots,p\}$ and $A \subseteq A_T^i$ for some T' in $\{1,\ldots,R_i\}$. If there exists $A_1 \in F_q$ such that $A_T^i \subset A_1$, then A_1 doesn't contain $B_1,\ldots,B_{q+1}$. Hence $A\ \ A_1$. This contradicts the definition of A. Hence $A_T^i \in K_{q+1}$. It is clear that A_T^i doesn't contain $B_1,\ldots,B_{q+1}$. By the definition of A we obtain $A=A_T^i$. The theorem is proved.

It can be seen that K and K^{-1} are determined uniquely by each other. Because of this fact, the determination of K^{-1} based on the algorithm doesn't depend on the order of sequence $\{B_1,\ldots,B_m\}$.

EXAMPLE: *Let* $X = \{1,2,3,4,5,6\}$ *and*
$K = \{(1,2),(2,3,4),(2,4,5),(4,6)\}$

According to the above algorithm we have:

$K_1 = \{(1,3,4,5,6),(2,3,4,5,6)\}$; $K_2 = \{(1,3,4,5,6),(2,3,5,6)(2,4,5,6)\}$

$K_3 = \{(1,3,4,5,6),(2,3,5,6),(2,4,6)\}$; $K_4 = \{(2,3,5,6)(1,3,4,5)(1,3,5,6),(2,4)\}$

$K^{-1} = K_4$.

We consider the following matrix:

The attributes: $\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$

$$M = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 \\ 0 & 3 & 0 & 3 & 0 & 0 \\ 4 & 0 & 4 & 0 & 4 & 4 \end{matrix}$$

M represents K, see [4].

Now we describe the "reverse" algorithm:
For given Sperner-system considered as the set of antikeys, we construct it's origin.

The following definition is necessary for us.
Let F be a closure operation over X. Denote:

$$Z(F) = \{A:F(A)=A\} \quad \text{and} \quad Y(F) = \{A\subset X:F(A)=A \quad \text{and} \quad \overline{\exists} B\in Z(F)\setminus\{X\}:A\subset B\}$$

The elements of $Z(F)$ are called closure sets. It is clear that $Y(F)$ is the family of maximal closure sets.

Now we prove the following lemma:

Lemma 4.3.: A is an antikey if and only if A is the maximal closure set. That is: $K_F^{-1} = Y(F)$.

Proof. Let A is an antikey and suppose that $A \subset F(A)$. Hence $F(F(A))=F(A)=X$. Consequently A is a key. This contradicts to $\forall B\in K_F: B\not\subseteq A$. If there is A' such that $A\subset A'$ and $A'\in Z(F)\setminus\{X\}$, then A' is a key. This contradicts to $A' \subset X$.

On the other side if A is a maximal closure set but there exists $B (B\in K_F)$ such that $B \subseteq A$, then $F(A)=X$. This contradicts to $A\subset X$. If $A\subset D$ $(D \subseteq X)$ then it is clear that $F(D)=X$ (because A is the maximal closure set). Consequently A is anantikey.

The lemma is proved.

An algorithm finding a minimal key:
Let H be the Sperner-system, $B \in H$ and $a \in X \backslash B$. Suppose that $B = \{b_1, \dots, b_m\}$. Let $G = \{B_T \in H : a \notin B_T\}$ and $T_o = B \cup \{a\}$

$$T_{q+1} = \begin{cases} T_q \backslash \{b_{q+1}\} & if\ \forall B_i \in H \backslash G : T_q \backslash \{b_{q+1}\} \not\subseteq B_i \\ T_q & otherwise \end{cases}$$

Theorem 4.4. If H is a set of antikeys, then $\{T_o, \dots, T_m\}$ are the keys and T_m is a minimal key.

Proof. By the remark 4.1. there is a closure operation F such that $H = K_F^{-1}$. We prove the theorem by the induction.
It is obvious that T_o is a key. If T_q is the key and $T_{q+1} = T_q$, then T_{q+1} is a key. If $T_{q+1} = T_q \backslash \{b_{q+1}\}$ and $F(T_{q+1}) \neq X$, then by lemma 4.3 there is $B_T \in H$ such that $F(T_{q+1}) \subseteq B_T$. Hence $T_{q+1} \subseteq B_T$. This constradicts to $\forall B_T \in H : T_{q+1} \not\subseteq B_T$. Consequently, T_{q+1} is a key.

Now suppose that A is a proper subset of T_m. If $a \notin A$, then clearly $F(A) \neq X$. If $a \in A$, then there is $b_q \in B$ such that $b_q \in T_m \backslash A$ $(1 \leq q)$. By the given algorithm there is $B_T \in H \backslash G$ such that $T_{q-1} \{b_q\} \subseteq B_T$. We obtain $A \subseteq T_m \backslash \{b_q\} \subseteq T_{q-1} \backslash \{b_q\} \subseteq B_T$ by $T_m \subseteq T_q$ $(0 \leq q \leq m-1)$. Hence $F(A) \neq X$. Consequently, T_m is a minimal key. The theorem is proved.

Remark 4.5:

- It is best to choose B such that $/B/$ is minimal.
- If there is B such that $\forall B_T \in H$ and $B_T \neq B : B \cap B_T = \emptyset$ then $a \cup b$ is a minimal key $(\forall b \in B)$
- If $X \backslash \bigcup_{B_T \in H} B_T \neq \emptyset$ then $a \in X \backslash \bigcup_{B_T \in H} B_T$ is a minimal key.
- Let $Y = \bigcup_{B_T \in H} B_T$ $(B_T \neq B)$. If $B \backslash Y \neq \emptyset$ then it is best to choose

$T_o = B \cap Y \cup \{a\} \cup \{b\}$ $(b \in B \backslash Y)$.

Remark 4.6: Let H be an arbitrary Sperner-system and $A \subset X$. We can give an algorithm (which is analogous to the above one) to decide whether A is or isn't a key. If A is the key, then this algorithm find one A' such that $A' \subseteq A$ and A' is a minimal key.

Basing on theorem 4.4. We can find the minimal keys in concrete cases.

In the paper [4] the equalitysets of the relation are defined: Let R be a relation and h_i, $h_T \in R$. Denote

$$E(h_i,h_T) = \{a \in X: h_i(a)=h_T(a)\} \qquad (i \neq T)$$

Remark 4.7. Let R be a relation over X. $R=\{h_1,\dots,h_m\}$. Let $E_{iT} = \{a \in X: h_i(a)=h_T(a)\}$ where $1 \leq i \leq m$, $1 \leq T \leq m$ and $i \neq T$. Denote $M = \{E_{iT}$: there isn't $E_{s\tau}$ such that $E_{iT} \subset E_{s\tau}\}$ practically, it is possible that there are many E_{iT} which equal to each other. We choose one E_{iT} from M. According to Lemma 4.3 it can be seen that M is the set of antikeys. Basing on the theorem 4.4. and the Remark 4.7 we find the minimal keys.

EXAMPLE. Let $X = \{1,2,3,4,5,6\}$ and

R be the relation:

0	1	0	0	1	0
1	0	1	0	0	1
2	0	0	1	2	2
0	1	2	2	0	3
3	2	1	0	0	0

It can be seen that $M = \{(1,2),(3,4,5),(4,6)\}$, where $E_{14} = \{1,2\}$, $E_{15} = \{4,6\}$ and $E_{25} = \{3,4,5\}$.

By the Theorem 4.4 and the Remark 4.5 it is clear that: $\{1,3\}$, $\{1,4\}$, $\{1,5\}$, $\{1,6\}$, $\{2,3\}$, $\{2,4\}$, $\{2,5\}$, $\{2,6\}$ are the minimal keys. We use the algorithm (Theorem 4.4) with $T_o = \{3,4,6\}$ and $T_o = \{4,5,6\}$. It can be seen that $\{3,6\}$ and $\{5,6\}$ are the minimal keys.

Let K be an arbitrary Sperner-system. The following theorem has been proved in [2].

Theorem 4.8. ([2]). $\binom{\Delta(K)}{2} \geq |K^{-1}| \geq \Delta(K)-1$.

Denote by $\binom{X}{k}$ the family of all k-element subsets of X. Let

$F_k(n) = max\ \{\Delta(K) : K \subseteq \binom{X}{k},\ |X| = n\}$

Theorem 4.9 ([5]).

$$F_k(n) \geq \sqrt{2}\ \binom{2k-2}{k-1}^{\frac{1}{2}\left[\frac{n}{2k-1}\right]}$$

We define the function $f_{2k-1} : N \to N$ (N-the set of natural numbers) in following way

$$f_{2k-1}(n) = \begin{cases} \binom{2k-1}{k-1}^{\frac{n}{2k-1}} & \text{if } n \equiv 0 \pmod{(2k-1)} \\ \binom{2k-1}{k-1}^{\left[\frac{n}{2k-1}\right]-1} \times \binom{2k-1+p}{k-1} & \text{if } n \equiv p \pmod{(2k-1)} \text{ and } 1 \leq p \leq k-1 \\ \binom{2k-1}{k-1}^{\left[\frac{n}{2k-1}\right]} \times \binom{p}{k-1} & \text{if } n \equiv p \pmod{(2k-1)} \text{ and } k \leq p \leq 2k-2 \end{cases}$$

and

$$f_{2k-2}(n) = \begin{cases} \binom{2k-2}{k-1}^{\frac{n}{2n-2}} & \text{if } n \equiv 0 \pmod{(2k-2)} \\ \binom{2k-2}{k-1}^{\left[\frac{n}{2n-1}\right]-1} \times \binom{2k-2+p}{k-1} & \text{if } n \equiv p \pmod{(2k-2)} \text{ and } 1 \leq p \leq k-1 \\ \binom{2k-2}{k-1}^{\left[\frac{n}{2n-1}\right]} \times \binom{p}{k-1} & \text{if } n \equiv p \pmod{(2k-2)} \text{ and } k \leq p \leq 2k-3 \end{cases}$$

It is clear that $2k-1$ and $2k-2 \leq n$

Take a partition $X = X_1 \cup \ldots \cup X_m \cup W$, where $m = \left[\frac{n}{2k-1}\right]$ and $|X_i| = 2k-1$ $(1 \leq i \leq m)$. Let

$$K = \{B : |B| = k,\ B \subseteq X_i,\ \forall_i\} \quad \text{if} \quad |W| = 0$$

$K = \{B: |B|=k,\ B \subseteq X_i\ (1 \le i \le m-1) \text{ and } B \subseteq X_m \cup W\}$ if $1 \le |W| \le k-1$

$K = \{B: |B|=k,\ B \subseteq X_i\ (1 \le i \le m) \text{ and } B \subseteq W\}$ if $k \le |W| \le 2k-2$

It is clear that $K^{-1} = \{A: |A \cap X_i|=k-1,\ \forall_i\}$ if $|W| = 0$.

$K^{-1} = \{A: |A \cap X_i| = k-1\ (1 \le i \le m-1) \text{ and } |A \cap (X_m \cup W)| = k-1\}$ if $1 \le |W| \le k-1$

$K^{-1} = \{A: |A \cap X_i| = k-1\ (1 \le i \le m) \text{ and } |A \cap W| = k-1\}$ if $k \le |W| \le 2k-2$

It can be seen that $f_{2k-1}(n) = |K^{-1}|$

By the analogous way we take a partition

$X = X_1 \cup \ldots \cup X_m \cup W$, where $m = [\frac{n}{2k-2}]$ and $|X_i| = 2k-2$

Let $K = \{B: |B|=k,\ B \subseteq X_i,\ \forall_i\}$ if $|W|=0$

$K = \{B: |B|=k,\ B \subseteq X_i\ (1 \le i \le m-1) \text{ and } B \subseteq X_m \cup W\}$ if $1 \le |W| \le k-1$

$K = \{B: |B|=k,\ B \subseteq X_i\ (1 \le i \le m) \text{ and } B \subseteq W\}$ if $k \le |W| \le 2k-3$

It is clear that $f_{2k-2}(n) = |K^{-1}|$ and $f_{2k-2}(n) \ge \binom{2k-2}{k-1}^{[\frac{n}{2k-2}]}$

Theorem 4.10. Let $X = \{1, \ldots, n\}$.

If $n \equiv 0$, (mod $(2k-2)(2k-1)$) then $f_{2k-1}(n) > f_{2k-2}(n)$

If we fix k, then $\lim\limits_{n\to\infty} \frac{f_{2k-2}(n)}{f_{2k-2}(n)} = \infty$

Proof. If $k=2$ then it is easy to prove that $\forall_n: f_3(n) \ge f_2(n)$.
If $n=6$ or $n \ge 8$ then $f_3(n) > f_2(n)$.

$$\text{Let} \quad F = \frac{\binom{2k-1}{k-1}^{\frac{n}{2k-1}}}{\binom{2k-2}{k-1}^{\frac{n}{2k-2}}} = \frac{\left(\frac{2k-1}{k}\right)^{\frac{n}{2k-1}}}{\binom{2k-2}{k-1}^{\frac{n}{(2k-2)(2k-1)}}}$$

It is known that $n! = \sqrt{2\pi n}\ (\frac{n}{e})^n\ e^{\frac{\theta n}{12n}}$, where $0 < \theta_n < 1$.

So
$$F \geq \frac{\left(\frac{2k-1}{k}\right)^{\frac{n}{2k-1}}}{\left(\frac{e^{\frac{\theta n}{12(2k-2)}}}{\sqrt{\pi(k-1)}}\right)^{\frac{n}{(2k-2)(2k-1)}} \times 2^{\frac{n}{2k-1}}} \geq \frac{\left(1-\frac{1}{2k}\right)^{\frac{n}{2k-1}}}{\left(\frac{e^{\frac{1}{24(k-1)}}}{\sqrt{\pi(k-1)}}\right)^{\frac{n}{(2k-2)(2k-2)}}} = E$$

$$\ln E = \frac{n}{2k-1}\left(\ln\left(1-\frac{1}{2k}\right)+\frac{1}{2k-2}\left(\frac{1}{2}\ln(\pi(k-1))-\frac{1}{24(k-2)}\right)\right) = T$$

$$T \geq \frac{n}{2k-1}\left(\frac{1}{2k-2}\left(\frac{1}{2}\ln(\pi(k-1))-\frac{1}{24(k-1)}\right)-\frac{1}{2k-1}\right) \text{ by } \left|\ln\left(1-\frac{1}{2k}\right)\right| \leq \frac{1}{2k-1}$$

It is clear that if $k=3$ then $\frac{1}{2k-2}\left(\frac{1}{2}\ln(\pi(k-1))-\frac{1}{24(k-1)}\right)-\frac{1}{2k-1}>0$ and for every $k\geq 4$: $\frac{1}{2}\ln(\pi(k-1))-\frac{1}{24(k-1)} > 1$. Hence $\frac{1}{2k-2}\left(\frac{1}{2}\ln(\pi(k-1))-\frac{1}{24(k-1)}\right)-\frac{1}{2k-1} > 0$. Consequently, if $n\equiv 0 \pmod{(2k-2)(2k-1)}$ then $f_{2k-1}(n)>f_{2k-2}(n)$.

Now let n be an arbitrary natural number and k fixed. It can be seen that there exists a number $M>0$ such that

$$\frac{\binom{2k-1+p}{k-1}}{\binom{2k-1}{k-1}^{1+\frac{p}{2k-1}}} < M, \quad \frac{\binom{p}{k-1}}{\binom{2k-1}{k-1}^{\frac{p}{2k-1}}} < M, \quad \frac{\binom{2k-2+p}{k-1}}{\binom{2k-2}{k-1}^{1+\frac{p}{2k-2}}} < M,$$

$$\frac{\binom{p}{k-1}}{\binom{2k-2}{k-1}^{\frac{p}{2k-2}}} < M.$$

It can be seen that $\ln E \underset{n\to\infty}{\to} \infty$. Hence $F \underset{n\to\infty}{\to} \infty$.

Consequently: $\frac{f_{2k-1}(n)}{f_{2k-2}(n)} \underset{n\to\infty}{\to} \infty$ (It is easily seen that $k=2$ is also true)

The theorem is proved.

On the basis of theorem 4.10 and theorem 4.8 it is clear that

$$F_k(n) \geq \sqrt{2\, f_{2k-1}(n)}\ .$$

§.5. THE GENERAL FUNCTIONAL DEPENDENCY

In the paper [6] the concept of the general functional dependency is defined.

Let $X = \{1,\ldots,n\}$, R be a relation over X.

$$h,h'\in R:\ t_i(h,h') = \begin{cases} 0 & \text{if } h(i)\neq h'(i) \\ 1 & \text{if } h(i)=h'(i) \end{cases}$$

Let $t(h,h')=(t_1(h,h'),\ldots,t_n(h,h'))$

We say that (f,g) is a functional dependency iff f,g are the Boolean function of n variables.

Let $R \models (f,g) \iff \forall h,h'\in R: ft(h,h')=1 \Rightarrow gt(h,h')=1$

Denote by F the set of the functional dependencies, $B(f,g) =\{R: R \models (f,g)\}$, *for* $Y \subseteq F$ *let* $B(Y) = \bigcap\limits_{(f,g)\in Y} B(f,g)$

Denote $Y \models (f,g)$ iff $B(Y) \subseteq B(f,g)$ and let $C(Y) =$ $= \{(f,g)\in F: Y\models (f,g)\}$.

We denote $f\leq f'$ iff $\forall t\in E_2^n: f(t)=1 \Rightarrow f'(t)=1$ and $Y (Y\subseteq F)$ is a closure set if $Y=C(Y)$.

Let Y be a closure set and

$MAX(Y) = \{(f',g')\in Y: g' = max(f),\ f'=min(g),\ (f,g)\in Y\}$

where $max(f) = \bigwedge\limits_{(f,g)\in Y} g$ *and* $min(g) = \bigvee\limits_{(f,g)\in Y} f$

Let $MIN(Y) = \{(f',g')\in Y: g'=min(f), f'=max(g),\ (f,g)\in Y\}$

where $min(f) = \bigvee\limits_{(f,g)\in Y} g$ *and* $max(g) = \bigwedge\limits_{(f,g)\in Y} f$

Theorem 5.1 ([6]). Let Y be a closure set. Then (f,g) is an element of Y if and only if there exists $(f',g')\in MAX(Y)$ such that $f\leq f'$ and $g'\leq g$.

Theorem 5.2. Let Y be a closure set. Then (f,g) is anelement of Y if and only if there exists $(f',g')\in MAX(Y)$ and $(f'',g'')\in MIN(Y)$ such that $f''\leq f\leq f'$ and $g'\leq g\leq g''$.

Proof. By the theorem 5.1. it is clear that we have only to prove: there is $(f'',g'')\in MIN(Y)$ such that $f''\leq f$ and $g\leq g''$.

Let $g''=min(f)$ and $f''=max(min(f))$. It is clear that $g\leq g''$ and we have $(f, min(f))\in Y$ by $Y\models(f, min(f))$.

Consequently, $max(min(f))\leq f$ by the definition of $MIN(Y)$.
It is clear that $min(max(min(f)))\leq min(f)$. It can be seen that $min(f)\leq min(max(min(f)))$ (by $(f,g) \models(max(min(f)),min(f))$.
Hence $min(f)=min(max(min(f)))$. We obtain
$(max(min(f)),min(f))\in MIN(Y)$ by the definition of $MIN(Y)$.
Hence $(f'',g'')\in MIN(Y)$ hold. The theorem is proved.

Finally, I express any decpest gratitude to Professor DR Demetrovics János for his help and encouragement.

REFERENCES

[1] W.W.Armstrong; Dependency Structures of Data base Relationships. Information Processing 74, North-Holland Publ. Co. (1974) 580-583.

[2] A.Békéssy, J.Demetrovics, L.Hannák, G.OH.Katona P. Frankl; On the number of maximal dependencies in data relation of fixed order. Discrete Math., 30 (1980) 83-88

[3] E.F.Codd; Relational model of data for large shazed data banks. Communications of the ACM, 13, (1970) 377-384.

[4] J.Demetrovics, Relációs adatmodell logikai és strukturális vizsgálata. MTA-SZTAKI Tanulmányok, Budapest, 114 (1980)

[5] J.Demetrovics, Z.Füredi, G.O.H.Katona; Minimum matrix representation of closure operations.
Preprint of the mathematical institute of the Hungarian academy of sciences Budapest, 12 (1983) 1

[6] Б. Тальхайм; Зависимости в реляционных структурах данных. ACTA CYBERNETICA, Szeged (1984)

ÖSSZEFOGLALÁS

MEGJEGYZÉSEK A LEZÁRÁSI OPERÁCIÓKHOZ

VU DUC THI

A dolgozatunkban a minimális kulcsok és antikulcsok és a lezárási operáció közötti kapcsolatot vizsgáljuk.

Р Е З Ю М Е

ЗАМЕЧАНИЯ ОБ ОПЕРАЦИЯХ ЗАМЫКАНИЯ

В настоящей работе изучается связь между минимальными ключами, антиключами и операциями замыкания.

Közlemények 30/1984 91-99

MAPPING TO STORAGE OF A NETWORK STRUCTURE

Dr. GY. MEZEI

National Technical Information Centre
and Library

The scope of this paper does not allow us to describe (see [1]) ideas about the whole mapping. So we are going to deal only with segments and area design. After the functional (see [2]) analysis of the normalized data model it is usual to have an entity-type directly transformed into a record-type, and relation-types into sets. Then for the sake of efficiency some of the recordtypes should be divided into disjoint (see [3]), segments (or groups according to the CODASYL DBTG's term) (see [4]), and in a next phase these segments should be melted together into areas.

Both these phases apply cluster analysis. Together with the former phase of designing a conceptual data model which forms homogenous clusters, these three phases may be seen as a three-level cluster analysis method. The first level is conceptual data model design the second is segments design and the third level is area design. In the following we will deal with the peculiarities of the second and third level. Similarly in the frames of another approach we can see segments and area design as a transformation of a starting cluster--structure into an object cluster-structure (see [5]).

1. PECULIARITIES OF THE TRANSFORMATION PROCESS

The transformation tries only to improve the chances of the physical DB design. That is why reducing space or improve availability and recovery (all of them are important performance measures) are not in the focus of this paper. But we concentrate to response time which is the most important factor concerning the users of an information system.

For this purpose as a performance measure the following formula (or a similar one) can be created

$$CF = \sum_{i=1}^{r} \left[S_i \sum_{k=1}^{m} (M\,(i,k) \quad K\,(i,k)) \cdot \delta_{ik} \right], \qquad \text{where}$$

$$\delta_{i,k} = \begin{cases} 1 \quad if \quad E_{SZ_k} \cap E_{f_i} \neq \emptyset & , \qquad \text{where} \\ 0 \quad else & \end{cases}$$

$F = \{f_1, f_2, \ldots, f_r\}$ is the set of functions managing the DB

and E_{f_i} (where $i=1,2,\ldots,r$) is the subset of the attributes belonging to the f_i-th function.

$SZ = \{SZ_1. SZ_2,\ldots,SZ_m\}$ is the set of the DB segments types and

SZ_k (where $k=1,2,\ldots,m$) is the group of attributes belonging to the k-th segment type.

$\exists[(SZ_k \cap SZ_\ell \neq \emptyset) \wedge (SZ_k \not\subseteq SZ_\ell) \wedge (SZ_\ell \not\subseteq SZ_k)]$, where $k \neq \ell$,

$SZ_k, \; SZ_\ell \in SZ$ and $\ell=1,2,\ldots,m$.

$K(i,k)$ is the access time of SZ_k on a mass storage device.

$M(i,k)$ is the time of data handling of SZ_k in the central memory.

S_i is the estimated relative frequency of f_i

There are several factors which affect response time.

Important ones:

- Type of access: it must be distinguished retrieval time and update time

- Mode of access: sequential, random etc.
- Query complexity
- Frequency of reference

File designers who know something about the expected factors may be able to design more effective file organizations. But more essential design factors to be aware of decisions which hardware and software products, specially DBMS-components are going to be used. Optimizing the function CF above (or a similar one) can be only if functions *M (i,k)* and *K (i,k)* are thoroughly known.
And in practice those formerly mentioned decisions sometimes are taken later than having started segments design. Furthermore unfortunately *M (i,k)* and *K (i,k)* functions can be estimated well only after having taken decisions on file structures in a later phase of the DB design. And just because of this nature of the mapping to storage of a network structure that is essentially a feedback-oriented task (see *Figure 1)* which there always must be enough opportunity for the correcting decisions of the data administrator in.

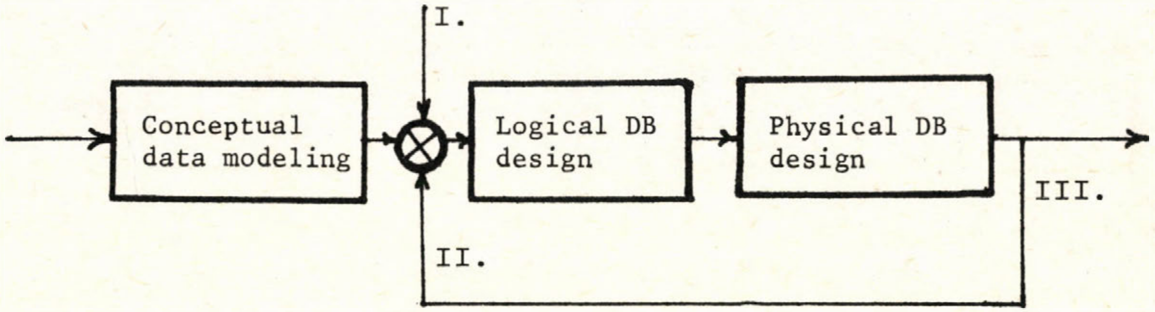


Figure 1

In the *Figure 1* two classes of entries can be seen.
Entry I: regeneration of the DB in as much as the information needs: - set of functions (accesses)
- frequency of functions
- priority of functions

or the hardware /software environment varies to a great extent.

Entry II: recombination (correction) of the structure of the segments since the decisions of the file organisation was not foreseeble.

The II-III. loop represents an iterative segments recombination process. Instead of using the rough model of *Figure 1.* there is use in avoiding its repetitive steps by splitting the logical design process into three consecutive tasks:

- segments design
- area design
- impact of DBMS applied

and modify the model of the process (see *Figure 2.*)

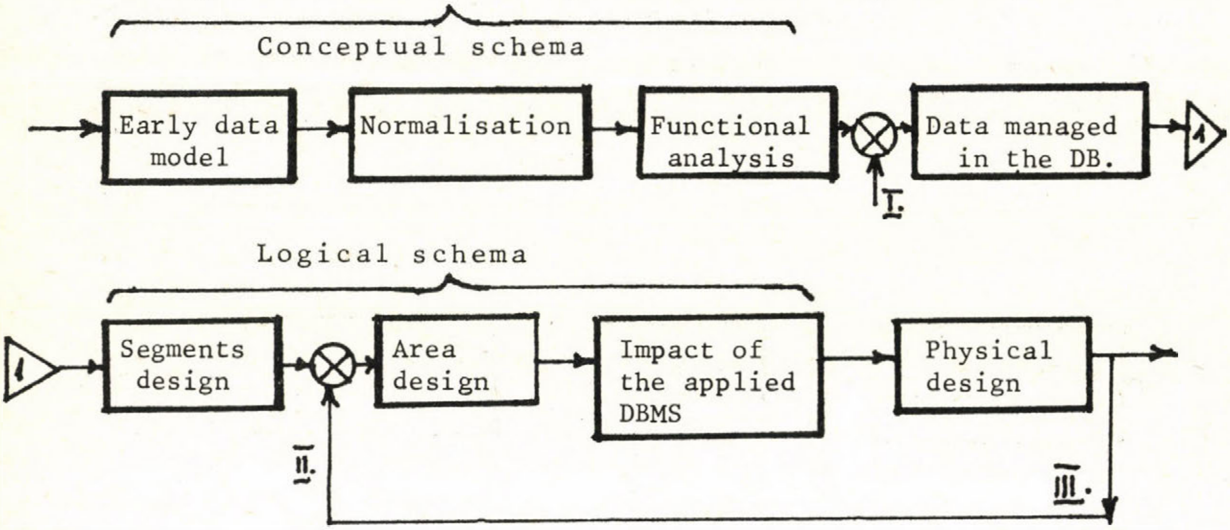


Figure 2.

If we apply that kind of segments-design method which is invariant (or nearly invariant) to small changes in the information needs of the organisation we may omitt segments-design from the loop.
So the data administrator has to frequently recombine the structure of the DB only at area level.
In the scope of 2. and 3. segments design and area design are discussed.

2. SEGMENTS DESIGN

2.1. THE STRUCTURE OF THE STARTING CLUSTERS

Before starting with segments we are given the product of the former process of data modeling. This product can be (see [6]) seen as a system of homogenous and generally overlapping clusters. The nucleus of such a cluster is the respective entity type and a cluster contains all the attributes related to that very entity type (that is why it is homogenous). The content and the number of these clusters is known.

2.2. CLUSTERING EFFICIENCY

In the case of a medium-size information system the number of the entity types is between 10-100 and that of the attributes (R) referring to one entity type is 10-30 (say R≈20), so the volume of the attributes altogether is generally some thousand, (say m=3000). It would be possible to cluster all the attributes together in one pass. But is well known that the bulk of the clustering methods is between $O(m \cdot \log m)$ and $O(m^2)$. So there seems to be use here in applying divide and conquer philosophy. It means that at a time the cluster analysis will be applied only for a homogenous subset of the attributes given by former data modeling (see 2.1).

By means of that simple trick the volume of attributes of larger systems and one pass clustering is transformed into a sequence of passes of clusterings dealing with moderate amounts of attributes. The number of the passes is equal to the number of entity types (i.e. the number of the formerly given clusters by data modeling).

2.3. THE FORESEEBLE STRUCTUREOF THE PRODUCED CLUSTERS

Segments design will produce an unforeseeble number of segmenttypes (clusters) the content of which is also unknown. Segmenttypes related to the same entity are disjoint ones. The segments (by definition) have an inner hierarchical structure. So in a clustering pass (see 2.2) we can make use of some kind of hierarchical clustering. Because of efficiency agglomerative methods seem to be advisable (see [8]).

Segmenttypes related to different entities might ovelap. This feature will be taken into consideration later only during area design (i.e. after having finished the sequence of the hierarchical dustering passes).

2.4. SELECTION OF THE HIERARCHICAL AGGLOMERATIVE CLUSTERING METHODS

2.4.1. Scale of variables

It determines to some extent the implementation of the hierarchical agglomerative clustering method. Stored similarity matrix approach is advisable, because of easy updating. Furthermore the DB managing functions (accesses) are seen as variables. The number of them is between N=100-1000. We concentrate only to the important ones, so $N \approx 100$. By weighting and standardising variables by the dmeanded estimated relative frequency of the accesses the scale of variables remain an interval one. But may be used subjective weighting as well and so the scale might become ordinal so stored similarity matrix is better (see [7]). This approach is effective when the number of the samples to be clustered (R) is less then the number of the (N) variables. That requirement is met in this case, since $R \approx 20$ and $n \approx 100$, so $R<N$ indeed. (see 2.2 as well).

2.4.2. Efficiency of the logical-physical design loop (see Figure 2)

In the class of hierarchical agglomerative clustering methods can be seen:
- linkage methods
- centroid methods
- variancia methods

Because for segments design such a method matches best which is invariant (or nearly invariant) to small changes in the information needs of the organisation, (see 1.) single-link methods are chosen (see [9]).
That subset of the single-link methods is preferable, which there is no need for cut-off level parameter and/or easy to program in.

2.4.3. Frequency of reference and mode of access

The relationship between the variables and the samples (here: attributes) reflects the frequency of reference for a sample (here: attribute).

To be frank sample is an unproper term here, because we can see attribute types here instead of samples. In a somewhat similar case of the leafs of the same tree where each occurence has got differences from the other ones probability based cluster analysis methods can be used. But in our case no such differences can be realized between the occurences of the same type. Furthermore the mode of access and the demanded subset of a particular attribute type in relation with a variable can be seen as a weighting factor of the type. Since the object-term matrix is essentially a non-binary one, the similarity coefficients (which are based on binary contigency tables) cannot be used here. This does not make good to the efficiency (space) of the segments design algorithm.

2.5. TAXONOMIC MEASURE

We can use only distance matrix with a distance measure which reflects asymmetry as well. The metric distance measure cannot reflect asymmetry so we choose a proper non-metric one. Because formely the Dice-coefficient was found as a proper similarity measure it comes handy Lance-Williams non-metric distance measure which is the inverse of the Dice-coefficient and can be used in the case of a non-binary object-term matrix (see [7]).

3. AREA DESIGN

3.1. THE STRUCTURE OF THE STARTING CLUSTERS

The structure of the starting clusters is written in 2.3. Useful additional information on conceptual data model is a list of those attributes which represent relations between two entity types.
It is important to know which segments contain these attributes. And from the point of view of implementation of the DB the distance measure components of these attributes should also be known. Furthermore necessary to be aware of the precedency (hierarchy) of the respective segment types when meeting the information requirement of each DB function.

3.2. THE FORESEEBLE STRUCTURE OF THE PRODUCED CLUSTERS

Area design is a necessary step in logical DB design, because (generally) none of the segment types can ensure the access of all the data required for a function of the organisation at a time. So between segment types either direct or indirect relations should be developed. Direct relations are developed first in the phase of area design. Are design will produce an unforeseeble number of area types (overlapping clusters). The content and the number of the elements (of these clusters) is also unknown.

3.3. THE TRANSFORMATION PROCESS

The transformation process should have the following features:

- harmonize clusters of objects (here: segment types) and clusters of variables at a time.
- easy to detect clusters by visualizing a display or hardcopy.
- quickly to select a representative subset of the segment types of each separate cluster.

Each representative subset determines an area. To meet the requirements above data-rearranging methods seen to be adequate.

LITERATURE

[1] Demetrovics, J., E. Knuth and P. Radó: Specification Meta Systems, IEEE 1982. May.

[2] Demetrovics, J., Gy. Gyepesi: Relációs adatmodell funkcionális függőségeinek általánosítása. MTA Alk. Mat. Lapok 6(1980) 313-322.

[3 CODASYL Systems Committee (1969): A Survey of Generalized DBMS May 1969. Report. New York.

[4] CODASYL systems Committee (1971): Feature Analysis of Generalized DBMS Technical Report. May 1971, New York - London - Amstardam

[5] Füstöss, A klaszteranalizis módszerei, MTA Szoc. Kut. Int. Módszertani füzetek (1977/1). Budapest.

[6] Halassy, B.: Adatmodellezés, adatbázis-tervezés, 1980. SZÁMOK, Budapest.

[7] Anderberg : Cluster Analysis for Applications
Academic Press Inc. New York - London, 1973.

[8] Van Rijsbergen: Information Retrieval. 1979.
Butterworths.

[9] Jardine - Sibson: Mathematical Taxonomy. 1971.
Wiley New York.

ÖSSZEFOGLALÁS

HÁLÓS ADATBÁZIS LEKÉPEZÉSE

Dr. Mezei Gyula

A cikk adatbázis szegmens- és area-tervezésével foglalkozik. A normalizált és funkcionálisan elemzett elvi adatmodell egyedtipusait rekordtipusokká, illetve részrekordokká /azaz szegmensekké/ képezik le, majd később e rekord /részrekord/ tipusokat nagyobb egységekbe /areak/ fogják össze. A cikk olyan módszert ismertet, ahol mindkét lépés során klaszternalizist használnak.

A szegmenstervezéshez agglomerativ hierarchikus klaszterálást és egy nem-metrikus távolságmértéket, az area- tervezéshez pedig táblázat-átrendező eljárást alkalmaznak.

ПРОЕКТИРОВАНИЕ СЕТЕВОЙ БАЗЫ ДАННЫХ

Д-р Дюла Мезеи

В статье рассматриваются вопросы проектирования сегментов и область /арэа/ базы данных. Отдельные типы нормализованной функциально проанализированной принципиальной модели данных преобразуются в рекорды либо в части рекордов /сегменты/, затем рекорды /части рекордов/ объединяются в большие группы /области/ /арэа/. Статья описывает также алгоритмы, которые используют методы кластерного анализа. Для построения сегментов использовались агглометративная иерархическая кластеризация и неметрическое измерение расстояний. При проектировании областей использовался метод перегруппировки таблиц.

Közlemények 31/1984 101-114

SHORT-TERM PRODUCTION AND DISTRIBUTION PLANNING OF STOCKPILING-DISTRIBUTION SUBSYSTEMS OF CRUDE OIL

P. INZELT and *B. UHRIN*
Computer and Automation Institute,
Hungarian Academy of Sciences

ABSTRACT

Stockpiling-distribution subsystems of crude oil products consist of a tankpark, connected together with a crude oil refinery. The system is in bilateral connection with other refineries, tank parks, consumer's terminals. The expected deliveries are forecast on the basis of the medium-term plan of the regional multirefinery system. An optimum short-term planning model of the technological operation to be performed in the subsystem is elaborated, considering the real technological restrictions. The simplified model of the subsystem is a large-size linear programming model.

INTRODUCTION

One of the prominent subsystems of a regional crude oil processing and oil-product stockpiling and distribution system is the high capacity tank park for the storage of crude oils, various intermediate feeds and products connected most frequently with oil refinery. Pipeline network and pumps within the tank park ensure the conveyance of the various materials to the technological units and between the tanks. Hereinafter the stockpiling-distribution system refers to the totality of the tank park and technological units of the refinery.

The system is in bilateral connection with the "outside world", i.e. with other refineries, tank parks, consumers. Crude oil, intermediate feeds (components), products may be transported from the outside into the system, or materials within the listed groups are transported out of the system. Transporation may take place by pipeline, railway, truck, barge.

The technological purpose of the system is outlined as follows:

- to ensure storage of the incoming crude oil, feeds, components and products
- to ensure shipment of the required feeds, components and products;
- to ensure storage of the feeds, components and products (buffer storage in conformity with the seasonal fluctuation of consumption);
- to ensure production of the specified part of the required products by blending from the components;
- to ensure production of the specified part of the components necessary for blending of the products; i.e. it is necessary to specify the quantity of the crude oil to be processed and the operating conditions of processing.

If the stockpiling-distribution system is regarded as the subsystem of a larger, regional system including several oil refineries and tank parks, then it is a metter of course that the proper knowledge of the productive capacity of each subsystem is highly significant under the given circumstances in respect of the optimal production planning of the total system. The production scheduling model to be presented was made with this purpose and as it will be shown at a later stage, the initial data were supplied by the medium term production planning of the regional system.

ASPECTS OF MODELLING

Given is the annual, or quarterly production plan of the regional, high level system. Naturally this includes the task of the individual refineries, tank parks (stockpiling-distribution systems) for the plan period in question. Task of the stockpiling-distribution system's production scheduling model is the following:

- with regard to the restrictions built into the model, to prepare the production program for the shorter periods in a way that the solution should meet the specifications of the production plan (search for the possible solution);
- to ensure the optimal functioning of the system according to given technical-economic objective function (search for the optimal solution);
- to take into consideration the stochastic character of the in- and out-bound deliveries at the specified materials of large volume and at the means of transporation. In present description only a very simple mode of depicting the random effects is being dealt with.

Essential requirement in connection with the model is independence from the concrete technological structure. The same model should be suitable for analysis of the various concrete stockpiling-distribution systems, furthermore for the analysis of the effect of the technological structure variation (e.g. analysing the effect of new investments or operating troubles).

In the following one of the possible, relatively simple varieties of the modelling and production scheduling of the stockpiling-distribution system including the technological units and tank park will be described. In the interest of reducing the dimensions of the task, the production scheduling is restricted to products of large volume (gasolines, motor fuels, fuel oils) and to their components, as well as to the crude oil utilization. Let us assume furthermore that the medium term production plan related to the system is known, i.e. the total quantity of the materials delivered into and transported out of the system is given for a fixed time interval.

DETERMINISTIC MODEL OF THE STOCKPILING-DISTRIBUTION SYSTEM

Let us regard the general stockpiling-distribution subsystem as a graph, the vertices of which are the tanks, while the edges are the pipelines connecting the tanks. For the purpose of general applicability, let us assume that every tank is connectible with every other one in two directions (technologically irrealistic connections are banned), material may arrive into every tak from outside of the system, and material maybe carried from every tank to the outside of the system. Content of the tanks is characterized with the most important quality parameters and every stocking is understood in term of blending, as a result of which "new" material will be stored is the tank.

The technological units are built in between specific tanks, tank groups, their function is regarded as separation or as specified alteration of the quality parameters. The material balance should be fulfilled for the separation-type processes, the lower-upper restrictions for the quantity of components derived in the process of separation, as well as the pertinent quality values are calculated in advance with the aid of the mathematical model of the technological unit. The given quality restrictions should be fulfilled during the process of blending.

Let us introduce the following notations:

$v_{ik}(t)$ - flow velocity of the material from tank-i to tank-k; $i, k=1,2,\dots,N;\ i\neq k;\ t\in[0,T];$

$v_i(t)$ - velocity of material flowing into tank-i from outside; $i=1,2,\dots,N;\ t\in[0,T];$

$w_i(t)$ - velocity of material carried from tank-i to outside; $i=1,2,\dots,N;\ t\in[0,T];$

V_i - capacity of tank-i; $i=1,2,\dots,N;$

$x_i(t)$ - quantity of material in tank-i; $i=1,2,\dots,N;$ $t\in[0,T];$

$\varphi_{ik}(t)$, $\psi_{ik}(t)$ - lower and upper restriction of $v_{ik}(t)$; $i, k=1,2,\ldots,N$; $i \neq k$; $t \in [0,T]$

ω_{ir}, η_{ir} - lower and upper restriction of the separation from tank-i; $i=1,\ldots,N$; $r=1,2,\ldots,R$;

$m_{ij}(t)$ - j-th quality index of the material in the tank-i $i=1,2,\ldots,N$; $j=1,2,\ldots,M$; $t \in [0,T]$;

$M_{ij}(t)$ - j-th quality index of the material delivered into tank-i from the outside; $i=1,2,\ldots,N$; $j=1,2,\ldots,M$; $t \in [0,T]$;

$\underline{m}_{ij}$, $\overline{m}_{ij}$ - lower and upper restriction of the j-th quality index of the material in tank-i; $i=1,2,\ldots,N$; $j=1,2,\ldots,M$;

$[0,T]$ - examined time interval

N - number of tanks

M - number of quality parameters

R - index occurring at separation (for interpretation, see later)

I, K_r - index sets occurring at separation (for interpretation, see later); $r=1,2,\ldots,R$.

In the course of setting up the model the trivial conditions are not detailed. Conditions of the model may be divided into two groups. The first one is a system of ordinary differential equations:

$$\frac{dx_i}{dt}(t) + \sum_{\substack{p=1\\p\neq i}}^{N} [v_{ip}(t) - v_i(t) + w_i(t)] = 0, \qquad (1a)$$

$$i=1,2,\ldots,N,$$

$$\frac{d(m_{ij}(t)\ x_i(t))}{dt} + \sum_{\substack{p=1 \\ p\neq i}}^{N} [m_{ij}(t)\ v_{ip}(t) - m_{pj}(t) v_{pi}(t)] +$$

$$+ m_{ij}(t) w_i(t) - M_{ij}(t) v_i(t) = 0, \qquad (1b)$$

$$i=1,2,\dots,N; \quad j=1,2,\dots,M.$$

The second group, includes the restrictions of the functions occurring in the differential equation system:

$$0 \leq x_i(t) \leq V_i, \qquad (2a)$$

$$i=1,2,\dots,N;$$

$$\varphi_{ik}(t) \leq v_{ik}(t) \leq \psi_{ik}(t), \qquad (2b)$$

$$k=1,2,\dots,N, \quad i\neq k,$$

$$\underline{m}_{ij} \leq m_{ij}(t) \leq \bar{m}_{ij}, \qquad (2c)$$

$$i=1,2,\dots,N; \quad j=1,2,\dots,M;$$

$$\omega_{ir} \leq \frac{\sum_{k\in K_r} v_{ik}(t)}{\sum_{\substack{p=1 \\ p\neq i}}^{N} v_{ip}(t)} \leq \eta_{ir} \qquad (2d)$$

$$i\in I\subset\{1,2,\dots,N\}, \quad r=1,2,\dots R.$$

Brief interpretation of the conditions is given, as follows:

(1a) - condition of conservation of matter (differential material balance)

(1b) - in case of assuming the blending according to linear relationship

(2a) - condition related to capacity of the tank

(2b) - restrictions of the flow velocities, these are generally the functions of the tank pipeline system and pumps

(2c) - following the blending process in the tanks, the lower, upper restrictions of the quality parameters of the product in the tank

(2d) - condition related to the separation type processes, where I represents those tanks, from which the separation takes place, $K_r, r=1,2,\ldots,R$ index set refers to the fact that after the separation type process the same product gets into the tanks pertaining to K_r. In case of given concrete system I and K_r are allocated in advance.

The outlined model is supplemented with certain special conditions necessary for the description of the separation type processes. Noteworthy is the following condition:

$$m_{pj}(t) = m_j(\omega_{pr}) + \frac{m_j(\eta_{pr})-m_j(\omega_{pr})}{\eta_{pr} - \omega_{pr}} \cdot \left[\frac{\sum\limits_{k\in K_r} v_{pk}(t)}{\sum\limits_{\substack{k=1\\k\neq p}}^{N} v_{pk}(t)} - \omega_{pr} \right], \quad \begin{array}{l} p\in I, \\ j=1,2,\ldots,M, \\ r=1,2,\ldots,R \end{array} \tag{3}$$

Here $m_j(\omega_{pr})$ and $m_j(\eta_{pr})$ are given constants. On the basis of the condition it is apparent that the separation type processes in the model are not characterized with discrete operation conditions, but the technological parameters influencing the operation are continuously variable between the physically determined lower and upper restrictions. Condition (3) expresses that the quality properties of the fractions derived in the process of separation are determined with linear interpolation based on the actual value of the quotient in relationship (2). The quality properties corresponding to the extreme values of the quotient are calculated in advance with the aid of the mathematical model of the technological unit.

In connection with condition (3) let us mention again that combination of the quality properties are expressed with linear approximating relationships. Naturally at certain concrete properties (e.g. flash point, viscosity) the special linearized relationships known from the literature are built into the model.

The *first* problem concerning the model, search of the *possible solution* can be outlined in the following way:

Given $x_i(t)$, $v_i(t)$, $w_i(t)$ and $m_{ij}(t)$ and the knowledge of the other parameters and functions figuring in the conditions, find non-negative functions $v_{ik}(t)$ fulfilling the conditions (1), (2), (3) in the interval $(t_1, t_2) \in [0, T]$.

The *second* problem is the following: such solution of the previous problem is to be found in the interval (t_1, t_2), which is optimal with respect to a linear objective function.

In the concluding section of the paper we shall return to the possible solution methods of the outlined two problems, i.e. the approximative solutions of the problems used by us will be briefly outlined. In the following part those more important random effects will be reviewed the consideration of which is advisable in the modelling process of the stockpilling-distribution system and the methodics applicable under our concrete circumstances will be described.

RANDOM EFFECTS IN THE STOCKPILING-DISTRIBUTION SYSTEM

The external random effects mentioned in the Introduction appear when the precise values of $v_i(t)$ and/or $w_i(t)$ are not known (at least for certain i-s), but they are random. By this the following is understood.

The in- or out-bound transportation of the materials (hereinafter movement) takes place in well separable charges, intermittently, while both the quantity of the material in motion and the length of time elapsed between completion of the previous movement and commencement of the next movement are random (random variables).

In first approximation let us assume that the random quantity of the material is normally distributed, while the random length of time is of exponential distribution.

With regard to those mentioned above, every single random $v_i(t)$ and/or $w_i(t)$ are described with the following type of "process":

Take $\xi_1, \xi_2, \ldots, \xi_m$ and $\eta_1, \eta_2, \ldots, \eta_m$ as the two series of random variables, where ξ_k is the random variable of the length of time in which movement k begins (calculated from completion of movement $k-1$), while η_k is the random variable of the total quantity of the material moving in movement k (charge). Thus, if t_{k-1} represents the moment of the completion of movement $k-1$, then $P(\tau_1 \leqq \xi_k < \tau_2)$ is the probability that movement k has not started until the moment $(t_{k-1} + \tau_1)$, but it starts off before moment $(t_{k-1} + \tau_2)$. Similarly $P(q_1 \leqq \eta_k < q_2)$ represents the probability that the total quantity of the material moving in movement k (charge) is between q_1 and q_2.

As noted above, in first approximation let us assume that ξ_k is of exponential distribution with expection and variance $1/\lambda_k$ $(\lambda_k > 0)$, i.e. its density function:

$$f_k(x) = \begin{cases} \lambda_k \cdot e^{-\lambda_k x} & x \geqq 0, \\ 0 & x < 0, \end{cases} \qquad (4)$$

$$k=1,2,\ldots,m,$$

Furthermore let us assume in first approximation that η_k is of normal distribution with expectation p_k and variance δ_k, the density function of which is:

$$g_k(x) = \frac{1}{\sqrt{2\pi}\ \delta_k} e^{-\frac{(x-p_k)^2}{2\delta_k^2}}, \qquad (5)$$

$$x \in R^1, \quad k=1,2,\ldots,m$$

On the basis of (4) and (5), $P(\tau_1 \leqq \xi_k < \tau_2)$, and $P(q_1 \leqq \eta_k < q_2)$ can be easily determined. ξ_k and η_k are generally not independent, their relationship should be examined in every practical case.

In the examined concrete system the probability that the movement begins depends not only on the time elapsed since the previous movement (this resulted in the exponential distribution), but also on the quantity of material still to be moved from the total quantity fixed by the production plan (of the material to be brought into motion in the whole [0,T] interval).

This effect was considered in first approximation by assuming that λ_k will depend also on the quantity of material still to be brought into motion. More precisely this means, that

$$\lambda_k = \Phi_k(X-D \cdot \sum_{i=1}^{k-1} p_k), \qquad k=1,2,\ldots,m \qquad (6)$$

(where D and X are constants).

Selection of the functions Φ_k depends on the practical case. Generally it can be stated that the less material quantity was moving during the previous $k-1$ movements, the time period, the movement k will start within, will be probably shorter, i.e. the higher is the argument of Φ_k, the lower is the expectation $1/\lambda_k$. Thus it is advisable to select a simple monotonously increasing function for the function Φ_k.

The fact that exactly quantity X will move for sure during the $[0,T]$ time, is expressed as

$$\sum_{k=1}^{m} p_k = X \qquad (7)$$

This refers also to the relationships among the λ_k-s.

Those described so far, give only the first approximation of the random effects occurring in our case.

This can be refined by use of other distributions expressing the reality better than (4) and (5) (e.g. "truncated" exponential, or β-distribution), using in place of (6) and (7) more precise mathematical description of the relationships among ξ_k-s and η_k-s, etc.

However, the most accurate description could be given by the thorough statistical analysis of the random effects, which has to be performed in every concrete case. The "right hand sides" of (1a), (1b) are obtained as a result of such analyses, and solution of the model is tackled only afterwards. This analysis is practically not possible within the technological system, but our model is functionally connectible with a simulation model, which can be used for the simulation of the random transportation into- and out of the system. This question here has not been dealt with.

The stochastic character of $v_i(t)$ and $w_i(t)$ entails not only the mentioned difficulties (namely the precise description of such random effects automatically causes great problems, as it was demonstrated previously), but it extremely aggravates the mathematical discussion and concrete solution of the model. This problem will be dealt with in the next section.

MATHEMATICAL AND COMPUTATIONAL REMARKS

Replacing the differential quotients in (1a) and (1b) by difference ones, we can express $w_i(t+\Delta t)$ and $m_{ij}(t+\Delta t)$ in terms of $v_{ip}(t)\cdot\Delta t$ ... etc. and $m_{ij}(t)\cdot v_{ip}(t)\cdot\Delta t$ ... etc. Taking into account that (2c) holds also for $m_{ij}(t+\Delta t)$, (1b) and (2c) can be approximately replaced by the following two inequalities:

$$[m_{ij}(t)-\underline{m}_{ij}]+[M_{ij}(t)-\underline{m}_{ij}]\cdot v_i(t)\cdot\Delta t-$$

$$-[m_{ij}(t)-\underline{m}_{ij}]\cdot w_i(t)\cdot\Delta t +$$

$$+\sum_{\substack{p=1\\p\neq i}}^{N}[m_{pj}(t)-\underline{m}_{ij}]\cdot v_{pi}(t)\cdot\Delta t - \sum_{\substack{p=1\\p\neq i}}^{N}[m_{ij}(t)-\underline{m}_{ij}]v_{ip}(t)\cdot\Delta t \geq 0, \tag{8}$$

and an analogous inequality with $\bar{m}_{ij}$.

Similarly, $x_i(t+\Delta t)$ has to fulfil (2a), hence (1a) and (2a) can also be replaced by two inequalities:

$$0 \leq x_i(t) + v_i(t)\cdot\Delta t - w_i(t)\Delta t + \sum_{\substack{p=1\\p\neq i}}^{N}[v_{pi}(t)-v_{ip}(t)]\cdot\Delta t \leq V_i. \tag{9}$$

Another possibility to eliminate the differential equations (1a), (1b) from the model (naturally only approximately), is to use (8), thus eliminating (1b), and after that replace the function $x_i(t)$ in (8) by an integral computed from (1a). (Naturally $x_i(t)$ too in (2a) has to be replaced by the integral.) In this case we obtain a system of linear inequalities where unknowns are the functions $v_{ik}(t)$ and their integrals

$$\int_{t_1}^{t} v_{ik}(\tau)\, d\tau.$$

This method is useful especially in the case when the interval (t_1, t_2) over which the model has to be solved, is small.

For solution of the system of linear inequalities many effective computing procedures and computer programmes are available. The programs usually give the so-called basic solution, where many $v_{ik}(t)$ are at zero level, which is quite reasonable from practical point of view.

The mentioned transformation of the model to a (dynamic) system of linear inequalities is also suitable, because the optimization turns now to a series of usual linear programming model(s). Here also many powerful computing packages exist. Use of the LP-technique has also other advantages, e.g. investigations of the sensitivity of the model, interpretation of the duality of LP, i.e. shadow prices, etc.

Also the random effects are more easily handled when we write the model in "linear inequality system" -form. In this case we are dealing with a system of "random linear inequalities". These systems are investigated in detail in stochastic programming (especially in the so-called "chance constrained programming"). The situation is now complicated by the fact that the problem is not a statistical but a dynamic one (i.e. it depends on time t). Hence the random effects are expressed as a t-parameter family of random variables and can be regarded as a general stochastic process.

It is necessary to note that treating of a system of random linear inequalities is after all an easier matter than to investigate a system of random differential equations.

ÖSSZEFOGLALÁS

KŐOLAJTERMÉKEK TERMELÉSI-FORGALMAZÁSI ALRENDSZERÉNEK RÖVID TÁVU TERMELÉS- ÉS ELOSZTÁS-TERVEZÉSÉNEK MODELLJE

Inzelt Péter - Uhrin Béla

Termelési-forgalmazási alrendszeren egy tank park valamint kőolaj-finomitók együttesét értjük. Az alrendszer kapcsolódik egyéb külső finomitókhoz, tank parkokhoz és fogyasztókhoz. Az alrendszerből történő ki- és be-szállitások várható összértékét ismertnek tekintjük. A modell a tank parkon belüli anyag-áramlást irja le valós technológiai korlátok figyelembevételével.

A modell célja az anyagáramlás optimális időbeli megadása, hogy a mindenkori ki- ill. be-szállitások teljesithetők ill. fogadhatók legyenek. A cikk a modell egyszerüsitését valamint bizonyos /külső/ véletlen effektusokat is tárgyalja.

МОДЕЛИРОВАНИЕ КРАТКОСРОЧНОГО ПЛАНИРОВАНИЯ ПРОДУКЦИИ И РАСПРЕДЕЛЕНИЯ В НЕФТЯНЫХ ПРОДУКЦИОННЫХ И СОХРАНЯЮЩИХ ПОДСИСТЕМАХ

П. Инзельт - Б. Ухрин

Цель модели, рассматриваемой в статье, заключается в планировании оптимального потока материалов между цистернами подсистемы, для выполнения данного годового плана. В статье разработаны методы упрощения модели /чтобы их можно было решить на ЭВМ/, а также влияние случайных эффектов.