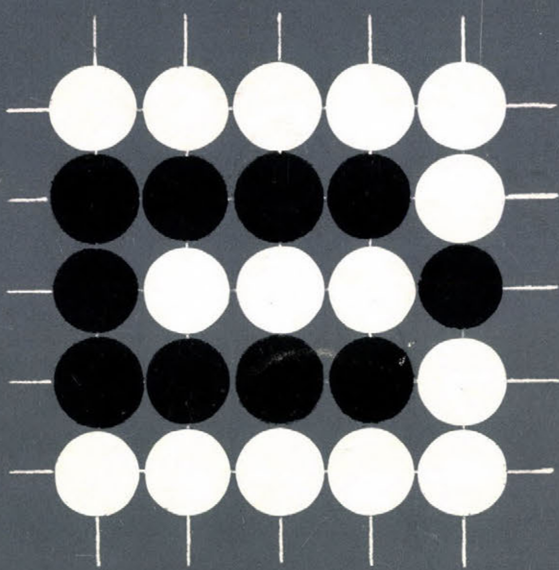
közlemények 24/1980

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

KÖZLEMÉNYEK

1980. JÚLIUS

Felelős kiadó:

DR VÁMOS TIBOR
igazgató

ISSN: 0324 – 2048

Készült a KSH SZÁMOK
nyomdájában

TARTALOMJEGYZÉK

CONTENTS

Procedings of the Computer and Automation Institute
Hungarian Academy of Sciences
Vol. 24.

СОДЕРЖАНИЕ

Труды Исследовательского Института
Вычислительной Техники и Автоматизации
Венгерской Академии наук
Выпуск 24.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények 24/1980.**

ГЕНЕРИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ И ИХ ПРЕДСТАВЛЕНИЕ С ПОМОЩЬЮ РЕЛЯЦИИ

Я. Деметрович, Дь. Дьепеши

Реляционная модель данных введенная Е.Ф. Коддом [6, 7], осуществляет хранение данных в форме матриц. Ряды этих матриц являются записями данных, столбцы - так называемыми <u>атрибутами</u>, т.е. теми свойствами, которые характеризуют отдельные данные.

Дадим теперь строгое определение.

Пусть $\Omega = \{a_1, \ldots, a_n\}$ непустое конечное множество. Конечное множество R называется <u>реляцией над Ω</u>, если элементы R являются функциями, определенными на Ω. Функции, составляющие множество R мы называем <u>строками R</u>.

Реляцию $R = \{f_1, \ldots, f_k\}$ над Ω можно представить в виде двухмерной таблицы:

	a_1	...	a_n
f_1	$f_1(a_1)$	...	$f_1(a_n)$
$\vdots$	$\vdots$	...	$\vdots$
f_k	$f_k(a_1)$	...	$f_k(a_n)$

В тесной связи с информационным обеспечением баз данных находится понятие <u>функциональной зависимости</u> [4,7]. Пусть R реляция над Ω и A, B - подмножества Ω. Мы говорим, что <u>B функционально зависит от A в</u> R ($A \xrightarrow[R]{f} B$), если для любых двух строк f и g реляции R :

$$(\forall a \in A)(f(a) = g(a)) \rightarrow (\forall b \in B)(f(b) = g(b)),$$

/т.е. если для каждого ряда R величины из B однозначно определяются величинами из A./

Обозначим через $\mathbf{F_R}$ множество функциональных зависимостей реляции R, т.е.

$$\mathbf{F_R} = \{(A,B) : A \frac{f}{R} B\}.$$

Если R реляция над Ω, то $\mathbf{F_R} \subseteq P(\Omega) \times P(\Omega)$. Назовем полными семействами такие подмножества $\mathbf{Y}$ множества $P(\Omega) \times P(\Omega)$, которые являются функциональными зависимостями некоторой реляции над Ω. Точнее, некоторое подмножество $\mathbf{Y} \subseteq P(\Omega) \times P(\Omega)$ является полным семейством, если существует реляция R над Ω, для которой $\mathbf{Y} = \mathbf{F_R}$.

В ходе исследования функциональных зависимостей первой задачей было абстрактное описание полных семейств. Впервые это удалось В.В. Армстронгу [1]. Он доказал, что некоторое $\mathbf{Y} \subseteq P(\Omega) \times P(\Omega)$ тогда и только тогда является полным семейством, когда для него выполняются следующие аксиомы: $(\forall A,B,C,D \subseteq \Omega)$:

(F1) $(A,A) \in \mathbf{Y}$;

(F2) $(A,B) \in \mathbf{Y}, (B,C) \in \mathbf{Y} \Rightarrow (A,C) \in \mathbf{Y}$;

(F3) $(A,B) \in \mathbf{Y}, C \supseteq A, D \subseteq B \Rightarrow (C,D) \in \mathbf{Y}$;

(F4) $(A,B) \in \mathbf{Y}, (C,D) \in \mathbf{Y} \Rightarrow (A \cup C, B \cup D) \in \mathbf{Y}$.

Совокупность аксиом (F1), (F2), (F3), (F4) мы называем f-аксиомами.

В этой статье мы занимаемся двумя проблемами теории полных семейств. Эти проблемы связаны со следующим вопросом: какое количество "информации" достаточно для описания некоторого полного семейства?

Дадим теперь строгую формулировку упомянутых проблем.

Проблема 1. Пусть $|\Omega| = n$. Каково наибольшее натуральное число $S(n)$, для которого справедливо следующее утверждение: существует такое полное семейство $F \subseteq P(\Omega) \times P(\Omega)$, что если R - реляция над Ω и $F_R = F$, то R содержит по крайней мере $S(n)$ строк?

Частный случай этой проблемы касающийся системы обозначенных ключей полного семейства, был рассмотрен в [9]:

Проблема 1'. Пусть $|\Omega| = n$. Каково наибольшее натуральное число $s(n)$, для которого справедливо следующее: существует такая система Шпернера $K \subseteq P(\Omega)$, что если R - реляция над Ω и $K = K(F_R)$, то R содержит по крайней мере $s(n)$ строк?

Проблема 2. Пусть $F \subseteq P(\Omega) \times P(\Omega)$ - полное семейство. Требуется охарактеризовать системы минимальной мощности, генерирующие F.

§1. Проблема 1. В этом параграфе будут даны оценки для $S(n)$ и $s(n)$. Проблема оценки $s(n)$ была рассмотрена в [9] где было доказано, что

$$\sqrt{2\binom{n}{[n/2]}} \leq s(n) \leq 2\binom{n}{[n/2]}.$$

В теореме 1.3 будет дано усиление этой оценки.

Дадим несколько определений.

Определение 1. Пусть $F \subseteq P(\Omega) \times P(\Omega)$ - полное семейство и пусть $(A, B) \in F$. В этом случае (A, B) называется максимальной зависимостью для F, если

$$B' \supseteq B,\ (A,B') \in F \rightarrow B' = B.$$

Через $M(F)$ будем обозначать множество максимальных зависимостей смейства F.

Множество $B \subseteq \Omega$ называется максимальной правой частью для F, если существует такое $A \subseteq \Omega$, что $(A, B) \in M(F)$. Через $I(F)$ будем обозначать множество максимальных правых частей семейства F.

Лемма 1.1. Пусть $F \subseteq P(\Omega) \times P(\Omega)$. Тогда

$$I(F) = \{B \subseteq \Omega : (\forall (A,C) \in F)(A \subseteq B \to C \subseteq B)\}.$$

Доказательство нетрудное, представляем его читателю.

Следствие 1.2. Пусть $F \subseteq P(\Omega) \times P(\Omega)$. Тогда $I(F)$ замкнуто относительно пересечения /т.е. $B, B' \in I(F) \to B \cap B' \in I(F)$/.

Доказательство. Согласно лемме 1.1,

$$I(F) = \{B \subseteq \Omega : (\forall (A,C) \in F)(A \subseteq B \to C \subseteq B)\}.$$

Пусть $B, B' \in I(F)$ и $(A, C) \in F$, тогда: если $A \subseteq B \cap B'$, то $A \subseteq B$ и $A \subseteq B'$, откуда $C \subseteq B$ и $C \subseteq B'$, т.е. $B \cap B' \in I(F)$.

В дальнейшем понадобится также

Определение 1.2. Пусть $F \subseteq P(\Omega) \times P(\Omega)$ и пусть $A \subseteq \Omega$; множество A называется ключом F, если $(A,\Omega) \in F$.

Множество A называется обозначенным ключом F, если A является ключом F и $A' \subseteq A, (A', \Omega) \in F \to A = A'$. Обозначим через $K(F)$ множество обозначенных ключей семейства F.

Легко заметить, что если K_i, K_j $(K_i \neq K_j) \in K(F)$, то $K_i \not\subset K_j$. Систему обладающей этим свойством назовем системой Шпернера.

Если F -полное семейство, то $K(F)$ является системой Шпернера и значит $|K(F)| \leq \binom{n}{[n/2]}$.

Теорема 1.3.

$$\frac{1}{n^2}\binom{n}{[n/2]} < s(n) \leq \binom{n}{[n/2]} + 1 .$$

Доказательство: Сначала покажем справедливость верхней оценки

Пусть $|\Omega| = n$ и $K \subseteq P(\Omega)$ - система Шпернера. Обозначим через L систему таких подмножеств Ω, которые не содержат ни одного элемента K и максимальны в этом смысле. Тогда L является системой Шпернера и, следовательно, $|L|=k\leqq\binom{n}{[n/2]}$. Пусть $L = \{X_2,\ldots,X_{k+1}\}$. Построим теперь реляцию R, содержащую $k+1$ строку : $h_1,\ldots, h_{k+1}$. Пусть $h_1(a) = 0$ для всех $a\in\Omega$ и пусть для $1< i \le k+1$

$$h_i(a) = \begin{cases} 0 \text{ , если } a\in X_i ; \\ i \text{ , если } a\notin X_i . \end{cases}$$

Если $A\in K$, то $A \not\subseteq X_i$ для любого i , и значит любые две строки реляции R отличаются друг от друга на множестве A, т.е. <u>A является ключом для F_R</u> .

Если $B\subset A$, то $B\notin K$, поскольку K - система Шпернера и B не содержит ни одного элемента K. Тогда существует такое i , что $B\subseteq X_i$. Отсюда следует, что h_0 и h_i совпадают на множестве B, так как они совпадают на X_i . С другой стороны $X_i \neq \Omega$, так как K $P(\Omega)$. По определению h_0 и h_i существует такое $a\in\Omega$, что $h_0(a)\neq h_i(a)$. Следовательно B не является ключом для F_R . Таким образом A является обозначенныи ключом для F_R .

Поскольку A было произвольным элементом множества K , то таким образом мы доказали, что $K \subseteq K(F_R)$.

Пусть теперь $B \in P(\Omega)\setminus K$, и предположим, что $B\in K(F_R)$ . . Поскольку $K\subseteq K(F_R)$, то B не содержит ни одного элемента K, и значит существует такое i , что $B\subseteq X_i$. Выше однако мы видели, что в этом случае B не является ключом для F_R , а следовательно, не может быть и обозначенным ключом F_R , что является противоречием.

Таким образом $K = K(F_R)$.

Приведем теперь доказательство справедливости нужной оценки, данное Л. Роняи.

Сначала сделаем два тривиальных замечания.

1. Если R-реляция, содержащая s строк, то существует реляция R', которая использует не более s символов и для которой $F_R = F_{R'}$ и $K(F_R) = K(F_{R'})$. /Нетрудно догадаться, что для F_R не имеет значения, совпадают ли или различаются символы, встречающиеся в различных столбцах реляции R. Поскольку в реляции R количество строк $\leq s$, то и число символов, встречающихся в одном столбце не превышает s./

2. Если R содержит s строк и $s' > s$, то существует такая реляция R', содержащая s' строк, что $F_R = F_{R'}$ /и конечно $K(F_R) = K(F_{R'})$ /.

Из утверждений 1 и 2 следует, что число систем Шпернера, каждая из которых является системой обозначенных ключей некоторой реляции, содержащей максимум $s(n)$ строк, - не превышает $(s(n))^{s(n)\cdot n}$. Таким образом $(s(n))^{s(n)\cdot n} > 2^{\binom{n}{[n/2]}}$, откуда путем несложных вычислений получаем, что

$$s(n) > \frac{1}{n^2} \binom{n}{[n/2]} \qquad \square$$

Следствие 1.4.

$$\frac{1}{n^2} \binom{n}{[n/2]} \leq S(n).$$

Теорема 1.5.

$$S(n) \leq \frac{3}{2} \binom{n}{[n/2]} + 1.$$

Доказательство. В.В. Армстронг [1] доказал, что если множество $I \subseteq \Omega$ замкнуто относительно пересечения, то существует только одно полное семейство F, для кторого $I = I(F)$.

Таким образом достаточно доказать, что если множество $I \subseteq \Omega$ замкнуто относительно пересечения, то существует реляция R, содержащая не более $\frac{3}{2} \binom{n}{[n/2]} + 1$ строк, для которой $I = I(F_R)$.

Пусть $I \subseteq \Omega$ замкнуто относительно пересечения. Пусть $N = \{Y \in I : Y \neq \cap\{Y' \in I : Y' \supset Y,\ Y' \neq Y\}\}$. Тогда N таково, что если $Y \neq Y' \in N$, то $Y \cap Y' \notin N$. Согласно теореме Д. Клейтмана [10], в этом случае $|N| = k \leq \frac{3}{2}\binom{n}{[n/2]}$. В лемме 2.1. будет доказано, что наименьшей системой, замкнутой относительно пересечения и содержащей N, является I. Значит достаточно построить такую реляцию R, которая содержит не более $\frac{3}{2}\binom{n}{[n/2]} + 1$ строк и для которой $N \subseteq I(F_R) \subseteq I$.

Пусть $N = \{Y_2, \ldots, Y_{k+1}\}$ и пусть R - та реляция, которую мы построили при доказательстве теоремы 1.3.; для $i = 2, \ldots, k+1$ X_i следует заменить на Y_i. Очевидно, что в этом случае

$N \subseteq I \subseteq I\ (F_R)$. □

§ 2. Проблема 2.

Перейдем теперь к Проблеме 2, которую впервые исследовал В.В. Армстронг [2]. Упомянем также, что с этой проблемой тесно связано понятие выводимости / см. [2], [11] /, а также исследование числа функциональных зависимостей, которые могут быть выведены из данной системы ключей [9].

Определение 2.1. Пусть $F \subseteq P(\Omega) \times P(\Omega)$ полное семейство. Мы говорим, что некоторое $F' \subseteq F$ генерирует F /или, что F' является генератором F /, если для реляции R над Ω, для которой $F' \subseteq F_R$, справедливо также, что $F \subseteq F_R$.

В [2] дана логическая характеристика систем минимальной мощности, генерирующих полные семейства. В теореме 2.2. мы дадим комбинаторный эквивалент этой характеристики, а также приведем оценку "минимальной мощности".

Определение 2.2. Пусть $F \subseteq P(\Omega) \times P(\Omega)$. Обозначим через $N(F)$ семейство элементов $I(F)$ замкнутое относительно пересечения, то есть $N(F)=\{Y \in I(F): Y \neq \cap\{Y' \in I(F): Y' \supset Y,\ Y' \neq Y\}\}$
Мы говорим, что $M \subseteq I(F)$ генерирует $I(F)$, если $I(F) = \{\cap M' : M' \subseteq M\}$.

Лемма 2.1. Подмножество $M \subseteq I(F)$ тогда и только тогда генерирует $I(F)$, когда $N(F) \subseteq M$.

Доказательство. Приводимое ниже доказательство хорошо известно в теории сетей.

Если M генерирует $I(F)$, то очевидно, что $N(F) \subseteq M$. Остается доказать, что $N(F)$ генерирует $I(F)$. Предположим, что это неверно. Пусть $X \in I(F)$ - множество наименьшей мощности, для которого $X \neq \cap\{Y \in N(F): Y \supseteq X\}$. Тогда разумеется, $X \notin N(F)$, то есть

$$(1) \quad X = \cap\{X' \in I(F): X' \supset X, X' \neq X\} .$$

Если $X' \supset X$, $X' \neq X$, то $|X'| > |X|$, и значит

$$(2) \quad X' = \cap\{Y \in N(F): Y \supseteq X'\} .$$

На основе (1) и (2) имеем:

$$X = \cap\{Y \in N(F): Y \supseteq X\} ,$$

что является противоречием. □

Теорема 2.2. Пусть F - полное семейство и $F' \subseteq F$. Утверждается, что F' тогда и только тогда является генератором F с минимальной мощностью, когда

$$(*) \quad (\forall Y \in N(F))(\exists A_Y \subseteq \Omega)(F' = \{(A_Y, Y) : Y \in N(F)\})$$

Доказательство. Если F' представимо в виде (*), то очевидно, что оно генерирует F.

Предположим теперь, что F' генерирует F. Для $(A,B) \in F'$ обозначим через B' максимальное множество, для которого $B' \supseteq B$ и $(A,B') \in F$. Тогда семейство $F'' = \{(A,B') : (A,B) \in F'\}$ также генерирует семейство F, причем $|F''| \leqq |F'|$. Очевидно, что множество $\{B' : \exists (A,B') \in F''\}$ генерирует $I(F)$, и значит по лемме 2.1

$$| N(F) | \leqq |F''| \leqq |F'| .$$

Далее, если $|N(F)| = |F''| = |F'|$, то

$$N(F) = \{B' : \exists (A,B') \in F'\} \qquad \square$$

Следствие 2.3. Если $F \subseteq P(\Omega) \times P(\Omega)$ - полное семейство и $|\Omega| = n$, то существует такое F', генерирующее F, для которого $|F'| \leqq \frac{3}{2} \binom{n}{[n/2]}$.

ЛИТЕРАТУРА

[1] ARMSTRONG, W.W.: Dependency structures of data base relationships, Information Processing 74, N.-H. Publ. Co.(1974), 580-583.

[2] ARMSTRONG, W.W.: On the generation of dependency stuctures of relational data bases, Publication 272, Université de Montréal (1977)

[3] ARMSTRONG, W.W., DELOBEL, C.: Decomposition and functional dependencies in relations, Publication 271, Departement D´ Informatique et de Recherche Operationelle, Université de Montréal (1979).

[4] BEERI, C., FAGIN, R., HOWARD, I.H.: A complete axiomatization for functional and multivalued dependencies in database relations, Proc. ACM SIGMOD Int. Conf. on Management of Data, Toronto, (1977), 47-61.

[5] BÉKÉSSY, A. - DEMETROVICS J.: Contribution to the theory of data base relations, Discrete Math. 27(1979), 1-10.

[6] CODD, E.F.: A relational model of data for large shared data banks, Comm. ACM, 13(1970) 377-387.

[7] CODD, E.F.: Further normalization of the data base relational model, Courant Computer Science Symposia 6 Data Base System, Prentice Hall, Englewood Cliffs, N.I. (1971) 33-64.

[8] CZÉDLI, G.: Függőségek relácios adatbázis modellben, Alk. Mat. Lapok (1980)

[9] DEMETROVICS, J.: Candidate keys and antichains, SIAM J. on Algebraic and Discrete Methods 1 (1980)

[10] KLEITMAN, D.: On a combinatorial problem of Erdős Proc. AMS (1966, February), 139-141.

[11] YU, C.T., JOHNSON, D.T.: On the complexity of finding the set of candidate keys for a given set of functional dependencies, Information Processing Letters, 5(1976), 100-101.

ÖSSZEFOGLALÁS

Funkcionális függőségek generálása és relációkkal való reprezentálása

J. Demetrovics - Gy. Gyepesi

Ebben a cikkben a funkcionális függőségek elméletének két kombinatórikus problémáját vizsgáljuk.

Először becslést adunk az [9]-ben definiált s(n)-re és S(n)-re, ahol s(n) az a legkisebb természetes szám, amelyre egy n elemü halmaz minden Sperner rendszere /részhalmazaiból álló antilánc/ reprezentálható s(n) soros reláció kulcsrendszereként; /S(n) ugyanez "Sperner rendszer" helyett "teljes család"-dal/.

Ezután kombinatórikusan jellemezzük a teljes családok minimális számosságu generátorrendszereit és becslést adunk erre a számosságra.

Generation of functional dependencies and their representation by relations

In this paper we deal with two combinatorial problems of the theory of functional dependencies.

Firstly we estimate s(n), defined in [9] and S(n), where s(n) is the minimal natural number such that every Sperner system of an n-element set may be represented as the set of candidata keys of a relation with s(n) rows; and S(n) is the same with "full fmaily" instead of "Sperner system".

Secondly we give a combinatorial characterization for the generating sets with minimal cardinality of full families and estimate this cardinality.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények 24/1980.**

О ВЫДЕЛИМЫХ МНОЖЕСТВ АРГУМЕНТОВ ФУНКЦИЙ

К.Н. Чимев

В работе исследуются вопросы, связанные с выделимыми множествами аргументов функций /выделимымыми m-торками/, структурными свойствами гиперграфов функций и инвариантностью выделимых m-торек функций при подстановке некоторых аргументов константами.

Используется терминология от [1 - 15] .

Теорема 1. Если функция $f(x_1,\ldots, x_n)$ порядка $n(n \geq 3)$ и x_1 образует выделимые m-торки для f только переменными $x_2,\ldots, x_r$ $(m \leq r \leq n-1)$, то для каждого $\ell (2 \leq \ell \leq m)$, каждая одна из переменных $x_{r+1},\ldots, x_n$ образует выделимую ℓ-торку для f, в которой участвует хоть одна из переменных $x_2,\ldots, x_r$.

Доказательство. С помощью теоремы 1 от [6] доказывается /при данных условиях/, что если предположим, что x_n не образует выделимую пару ни с одной из переменных $x_2,\ldots, x_n$, то существуют значения $c'_2,\ldots, c'_r$ для $x_2,\ldots, x_r$ таких, что функция

$$f_1= f(x_1,c'_2,\ldots, c'_r, x_{r+1},\ldots, x_n)$$

зависит существенно от x_1 и x_n.

Если число существенных переменных функции f_1 не больше m, из теоремы 4 от [7] следует, что существет выделимая m-торка для f, в которой участвуют x_1 и x_n , что невозможно.

Если число существенных переменных функции f_1 больше m, то из теоремы 1 от [5] следует, что существует выделимая m-торка для f, в которой участвует x_1 и хотя бы одна из переменных $x_{r+1}, \ldots, x_n$, что невозможно.

Из доказанного следует, что переменная x_n образует выделимую пару для f хотя бы с одной из переменных $x_2, \ldots, x_r$. Тогда из теоремы 4 от [7] следует, что для каждого ℓ, существует выделимая ℓ-торка для f, в которой участвует x_n и хотя бы одна из переменных $x_2, \ldots, x_r$.

По аналогичному способу теорема 1 доказывается и для каждой одной из переменных $x_{r+1}, \ldots, x_{n-1}$.

Нужно отметить, что при $m = \ell = 2$ утверждение в теореме 1 доказано в [8].

Гиперграф функции $f(x_1, \ldots, x_n)$, по отношении выделимы m-торек функции, будем называть гиперграфом с вершинами существенные переменные функции f и ребрами - выделимые m-торек функции f.

Гиперграф функции f по отношении выделимые m-торек будем обозначать с H_f^m.

Две вершины гиперграфа будем называть смежными, если существует ребро, которое содержит эти вершины.

Если две вершины гиперграфа смежные, будем говорить, что расстояние между ними равно 1.

Если вершины V_1 и V_2 гиперграфа не смежные, но существует вершина V_3, которая смежная с каждым из вершин V_1 и V_2, будем говорить, что расстояние между V_1 и V_2 равно 2.

<u>Теорема 2</u>. Если функция $f(x_1, \ldots, x_n)$ порядка $n (n \geq 3)$, то для каждого $m (2 \leq m \leq n)$ гиперграфа H_f^m связаны и расстояние

между какими-нибудь двумя вершинами не превосходит 2.

Верность теоремы следует из теоремы 1.

Теорема 3. Если функция $f(x_1,\dots,x_n)$ порядка $n (n \geqq 3)$, и переменная x_1 образует выделимые m-торек для f только с переменными $x_2,\dots, x_r (m \leqq r \leqq n-1)$, то для каждого натурального числа $\ell (2 \leqq \ell \leqq m)$, переменная x_1 не образует выделимые ℓ-торек, в которых участвуют переменные из множества $\{x_{r+1},\dots, x_n\}$.

Теорему 3 можно доказать с помощью теоремы 4 от [7] .

Следствие. При условиях теоремы 3 для каждого знаяения c_1 для x_1 функция $f(x_1 = c_1)$ зависит существенно от переменных $x_{r+1},\dots, x_n$.

Верность утверждения следует из теоремы 3 и теоремы 1 от [6].

Теорема 4. Если функция $f(x_1,\dots,x_n)$ порядка $n (n \geqq 3)$ и переменная x_1 образует только одну ыделимую r-торку для f с переменными $x_2,\dots,x_r (2 \leqq r \leqq n-1)$, то для каждого $i = 2,\dots, r$, пара (x_1, x_i) выделима для f.

Доказательство. При условиях теоремы, не ограничивая общности рассуждения, докажем, например, что пара (x_1, x_2), выделима для $f(x_1,\dots, x_n)$.

Допустим, что пара (x_1, x_2) не выделимая для $f(x_1,\dots, x_n)$. Тогда с помощью теоремы 3 и теоремы 1 от [6] доказывается, что существует функция g порядка ℓ, $2 \leqq \ell \leqq r$ которая зависит существенно от x_1 и x_n. Тогда из теоремы 4 от [7] следует, что существует выделимая r-торка для f, в которой участвуют x_1 и x_n, что невозможно.

Теорема 5. Пусть $f(x_1,\dots, x_n)$ функция порядка $n (n \geqslant 3)$ и x_1

образует выделимые m-торек для f только с переменными $x_2, \ldots, x_r$ $(m \leqq r \leqq n-1)$.

Если одна ℓ-торка $(2 \leqq \ell \leqq m)$ выделима для f и содержит хотя бы одну из переменных $x_{r+1}, \ldots, x_n$, то она выделима и для каждой одной, из функций $f(x_1 = c_1)$, где c_1 какое-нибудь значение для x_1.

Доказательство. Теорему 5 можно доказать отдельно для $\ell = m$ и отдельно для $2 \leqq \ell \leqq m-1$. При этом можно воспользоваться теоремами 3 и 4, 1 от [5] , 1 от 6 и 4 от [7].

Будем говорить, что переменная x_i порядка p для функции f, по отношению выделимых m-торек, если существуют точно p выделимые m-торек для f, в которых участвует x_i.

Порядка переменная x_i для функции f, по отношении выделимых m-торек будем обозначать с $P^m_{x_i,f}$

Теорема 6. Если не существует выделимая m-торка для функции $f(x_1, \ldots, x_n)$, которая содержит переменные x_i и x_j, то

$$P^m_{x_i,f} = P^m_{x_i,f\,(x_j = c_j)},$$

где c_j какое-нибудь значение для x_j.

Верность утверждения в теореме следует из теоремы 5.

Теорема 6 является обобщением теоремы 1 от [6] .

Следствие. Если вершины x_i и x_j не смежные в гиперграфе H^m_f , то какое бы ни было значение c_j для x_j, степень вершины x_i для гиперграфа $H^m_{f(x_j = c_j)}$ равна степени вершины x_i для гиперграфа H^m_f .

Л И Т Е Р А Т У Р А

1. Яблонский С.В.: Функциональные построения в k-значной логике. Труды математического института им. В.А. Стеклова, т.51, 1958, 5-142.

2. Яблонский С.В., Гаврилов Г.П., Кудрявцев В.Б.: Функции алгебры логики и классы Поста, Москва, 1966.

3. Schwartz R.E.: Existence and uniquenese properties of boolean functions. SIAM Jurnal on applied matematics, 1970, 18, 2, 454-461.

4. Тоом А.Л.: О сложности реализации двоичных функций, имеющих мало "подфункции". Пробл. кибернетики, вып. 18, 1967, 83-90.

5. Чимев К.Н.: Върху някои свойства на функциите, год. на ВТУЗ. Математика, т.УII, кн. 1, 1971 23-32.

6. Чимев К.Н.: Върху инвариантността на отделимите двойки на функциите. Год. на ВТУЗ. Математика, т. УIII, кн. 1, 1972, 129-136.

7. Чимев К.Н.: Върху подфункциите и силно съществените променливи на функциите. Год. на ВТУЗ. Математика, т.IX, кн. 4, 1973, 43-55.

8. Чимев К.Н.: Върху отделимите двойки на функциите. Год. на ВТУЗ, Математика, т. УII, кн. 3, 1971, 7-12.

9. Лупанов О.Б.: Об одном классе схем из функциональных элементов. Сб. "Проблемы кибернетики", вып. 7, 1962, 61-114.

10. Соловьев Н.А.: К вопросу о существенной зависимости функции алгебры логики. Сб. "Проблемы кибернетики", вып.9, 1963, 333-335.

11. Бейгбарт Ю.Я.: О существенных переменных функции алгебры логики. ДАН СССР, 1967, т. 172, № 1, 9-10.

12. Solomaa A.: On essential variables of functions, especially in the algebra of logic. Annales academiae scientiarum fennicae, ser. A, 339 /1963/, 1-11.

13. Чимев К.Н.: Върху силно съществените промениливи на функциите от P_k. Год. на ВТУЗ. Математика, т.У, кн. 2, 1968/69, 155-162.

14. Чимев К.Н.: Върху зависимостта на функциите от P_k от аргументите им. Год на ВТУЗ. Математика, т. IУ, кн 3, 1967, 5-13.

15. Чимев К.Н.: Върху отделимите подмножества и сильно съществените променливи на функциите. Год. на ВУЗ. Приложна математика, т. X, кн. 4, 1974, 7-13.

ÖSSZEFOGLALÁS

Függvény argumentumai halmazának leválasztásáról

K. N. Csimev

A szerző hat tételt bizonyit be függvény argumentumai halmazainak leválasztásáról, függvény hipergráfjainak strukturájáról és függvények leválasztott m-tóruszocskáinak konstans helyettesitéssel szembeni invarianciájáról.

On selection of argument sets

K. N. Tshimev

In the paper six theorems will be proved about the selection of argument sets from a function, about the hypergraphs' structure of a function and about the invariance of the "m-tuple" with respect to the substitution of some arguments.

MTA Számitástechnikai és Automatizálási Kutató Intézete, Közlemények 24/1980.

ON A CERTAIN DISTRIBUTED DATA BASE MODEL

Józef Kubit
The Academy of Economics in Cracow
Department of Computer Science
Poland

ABSTRACT

The paper presents a formal model of distributed data base. Present work is an attempt at defining fundamental notion connceted with decompositions of data base model. Distributed data base model consists of family of individual (local) data base models and administrator of distributed data base model. Model is based on a definition of relational schema of data. Data are the object of widely understood measurement.

1. INTRODUCTION

In recent yeast many different models have been presented describe the data base at the conceptual (mathematical, logical) level. The rapid development of large computer networks led to investigations about distributing data base throughout such a system. Looking at these trends we can identify two basic approaches [Neuhold 1977]:

1) Using already existing networks of large computer systems,
2) As an alternatice, we may build a new data base management system which takes into account the existence of a computer network but does not rely on already existing indivudual (local) data base systems.

We envision for thenext future, that computer networks commonly will encompass both large minicomputers even including inteligent terminals.

The reason for the interest in distributed data base systems is because the provide a solution to very real problems for the geographically distributed organization which needs to preserve a unified information--sharing and processing system.

* The paper has been presented on The International Colloquium COMPCONTROL'79 in Sopron on November, 1979.

In general, the advantage of centralized approach are the disadvantages of a distributed approach and vice versa.
They are, in broad form [Champine 1977]:
Distributed advantages /centralized disadvantages

- communication failsoft capability
- lower communications data rate and cost
- configuration flexibility
- high system performance (fast response and high transition rate)
- modular implementation
- modulat up-grade

Generalized advantages / distributed disadvantages

- operations economy
- hardware economy of scale
- unified control
- easy updata / retrieval
- compatibility.

Gradually along with a vigorous development of data base systems based on Codd's relational model of data, a new theory of relations has emerged. The birth of this theory must be assigned to Cood's papers [1970] and [1971] where the basic notions were introduced and collected together.

Depending on delimited purposes of realized work the notion of data base model is not used similarly. It concerns for example the works devoted theory of i.s.r systems [Marek and Pawlak 1976].

General comparison of Codd's and Pawlak's and Marek's approach was done by Koczkodaj [1978] who showed that they lead to essentially same consequences.

Many data models have already been proposed, wach having its own concepts and terminology. In studying the various models is becomes apparent that they have similarities and differences which are not trivial to analyze [Kerschberg, Klug and Tsichritzis 1976].

The notion of data base model is connected with the notion of information system model. Information system model in [Kubit 1979] consists of four specialized system models. Two of them constitute a certain data base model.

Information system model is component of controlled economic system model.

The purpose of the paper is a presentation of divided - distributed data base model.

In each of distributed data base cases we deal as follows:

The base B is decomposed into smaller entities $B_1, \dots, B_i, \dots, B_n$ and we add new object called administrator ($\bar{A}$). The administrator of queries. Same of i'S could be empty.
The response for i is the settheoretical union of the responses of B_i to i_i.

Of particular importance may be a formulation of relations occuring between notions of data base model and administrator of data base model.

2. NOTION OF DATA

Let's take an attribute names $\mathbf{N}$. This set will be named alphabet of data names. From elements of this alphabet we'll make a word set $\mathbf{N}^*$.

Let N be a subset of $\mathbf{N}^*$ set. N set will be named a set of data names. Data names will be marked by N while for all N there will be

$$N = < n_1, \dots, n_j, \dots, n_{r_N} > \quad n_j \in \mathbf{N}$$

Denote $<n_1, \dots, n_j, \dots, n_{r_N}>$ will be also defined relational schema of data.

Let's take a nonempty family of domains $\{M_i\}_{i \in I}$ and mapping m.
The mapping m need not be one-to-one. This is onto mapping

$$m: \ \mathbf{N} \twoheadrightarrow \{M_i\}_{i \in I}$$

According to what was said above for relational schema of data will correspond a finite relation

$$R_N = m(n_1) \mathbf{X} \dots \mathbf{X} m(n_j) \mathbf{X} \dots \mathbf{X} m(n_{r_N})$$

The set of data values corresponding relational schema N will be defined as $\gamma(N)$, $\gamma(N)=\mathbf{P}(R_N)$ where $\mathbf{P}(R_N)$ R_N.

By V will be denoted an element of subset $\mathbf{P}(R_N)$, $V \in \mathbf{P}(R_N)$. Element V will be named data value and

$$V = \langle v_1, \ldots, v_j, \ldots, v_{r_N} \rangle$$

DEFINITION 1

Data d will be named a pair composed of data name and data value $d = \langle N, V \rangle$.

Names n_j are the names of specified attributes, whereas parts v_j of data value V are the values of these attributes. Attributes which is defined by n_j includes the definition of scale and that of value unit v_j. Further on will be identified as equivalence class of data $\mathbf{d} = \| d \|$.

Data will be obtained in the process of measurement in the broadest sense of the notion. Data measurement inotion will be defined by means of data measurement function. Data measurement function will be referred to as isomorphism from real objects (portions of productive factors) algebra, onto data algebra

$$h: \mathbf{A} \rightarrowtail\!\!\twoheadrightarrow \mathbf{D}$$

where: A - universe of algebra of equivalence classes of real objects $\mathbf{A} = A\langle O_1, \ldots, O_n \rangle$

D - universe of data algebra $\mathbf{D} = \langle D, O'_1, \ldots, O'_n \rangle$ (cf. [Kobayashi 1975])

Dated $\mathbf{d} \in D$ are data equivalence classes because of defined equivalence relation. Algebras $\mathbf{A}$ and $\mathbf{D}$ are similar algebras. For these two algebras the following theorem holds true.

Theorem 1

Let f be one-to-one transformation of A_0 set of algebra generators onto D_0 set of similar algebra **D** generators. If there exists homomorphism h of algebra **A** into algebra **D** being an extension f and also homomorphosm g of algebra **D** into algebra **A** being an extension f^{-1}, then h is isomorphism **A** onto **D** and $h^{-1}=g$.

3. SYSTEM OF DATA RETRIEVAL

We assume that I_N denotes a set of attribute names according to the relational schema of data (data name) N. By m_N will be understood restriction of m function to I_N set of attribute names

$$m_N = M \uparrow I_N$$

Let $\bar{U}$ be a set of all functions m_N for set **N** of data names

$$\bar{U} = \{m_N\}_{N \in \mathbf{N}}$$

Let I_N^* denote a set of all words overthethe alphabet I_N of attribute names of data N. The elements of I_N set will be marked as $\bar{n}$.

By $\mathbf{P}(R_N)[\bar{n}]$ we shall define a set of all data value V restrictions to a set of attribute names $\bar{n}$ according to a relational schema of data name N

$$\mathbf{P}R_N[\bar{n}] = \{V\,\bar{n} : V \in \mathbf{P}(R_N)\}$$

Let $\mathbf{P}(R_N)^*$ be understood as a set of all projections $\mathbf{P}(R_N)$ for all $\bar{n} \in I_N^*$. By **P** we shall mark a set of all projections $\mathbf{P}(R_N)^*$ for each data name N

$$= \{(R_N)^*\}_{N \in \mathbf{N}}$$

We assume the Θ_N^j will denote a set of two-argument order relations j-type values domain assigned to the data name N. Two-argument order relations Θ_N^j give a set Θ of order relations of data name attribute values which is expressed in the following way

$$\Theta = \{\{\Theta_N^j\}_{j \in \{1,\ldots,r_N\}}\}_{N \in \mathcal{N}}$$

By Q we shall mark a set of set theory operations [Neuhold 1974] defined for **P** set.

DEFINITION 2

System of data retrieval for D set of data will be marked seventuple

$$S_{DR} = \langle D, N, N, \bar{U}, \ , \Theta, Q\rangle$$

where:

D - universe of data algebra

N - alphabet of data names

N - set of data names

$\bar{U}$ - set of functions m_N (assigning attribute values domains) for data defined by relational schema from set

P - set of all projections of data values from data set D

Θ - set of two-argument order relations of attribute values

Q - set of set theory operations for set of data values projections.

4. SYSTEM OF CONTROL SYSTEM SETS OF DATA RETRIEVAL

Let set N be alphabet of subset of data names for set D. From elements N of set N we compose of set of all words N^*. Let $\hat{N}$ be a subset of N^* set. $\hat{N}$ will be names a set of names of data subsets. The names of data subsets will be expressed by $\hat{N}$, where for each $\hat{N}$ there will be

$$\hat{N} = \langle N_1, \ldots, N_k, \ldots, N_{w_N}\rangle ; \qquad N_k \in N$$

The denote $\hat{N}$ will be defined as a name of subset or relational schema of data subsets (sets).

We say that $\hat{N}$ is a lattice with respect of the operation $\cup$ (join) and $\cap$ (meet) if the following equations hold (axioms of lattice) (cf. [Kuratowski and Mostowski 1976]).

$$(1) \quad \hat{N} \cup \hat{N} = \hat{N} \qquad\qquad \hat{N} \quad \hat{N} = \hat{N}$$

$$(2) \quad \hat{N} \cup \hat{N}' = \hat{N}' \cup \hat{N} \qquad\qquad \hat{N} \cap \hat{N}' = \hat{N}' \cap \hat{N}$$

$$(3) \quad \hat{N} \cup (\hat{N}' \cup \hat{N}'') = (\hat{N} \cup \hat{N}') \cup \hat{N}'' \qquad\qquad \hat{N} \quad \hat{N} \quad \hat{N} = \hat{N} \quad \hat{N} \quad \hat{N}$$

$$(4) \quad \hat{N} \cap (\hat{N} \cup \hat{N}') = \hat{N} \qquad\qquad \hat{N} \cup (\hat{N} \cap \hat{N}') = \hat{N}$$

We call a lattice distributive if

$$(5) \quad \hat{N} \cap (\hat{N}' \cup \hat{N}'') = (\hat{N} \cap \hat{N}') \cup (\hat{N} \cap \hat{N}'') \qquad\qquad \hat{N} \cup (\hat{N}' \cap \hat{N}'') = (\hat{N} \cup \hat{N}') \quad (\hat{N} \cup \hat{N}'')$$

We introduce an order relation between elements of lattice:

$$(6) \quad \hat{N} \leqq \hat{N}' \equiv \hat{N} \cup \hat{N}' = \hat{N}'$$

or, equivalenty,

$$(7) \quad \hat{N} \leqq \hat{N}' \equiv \hat{N} \cap \hat{N}' = \hat{N}$$

Similarly we define the elements o and i as the elements satisfying conditions

$$(8) \quad \hat{N} \cup o = \hat{N}, \qquad\qquad \hat{N} \cap i = \hat{N}$$

It is easy to show o is the smallest element in $\hat{N}$ and that i is the largest, namely for every $\hat{N} \in \hat{N}$

$$(9) \quad o \leqq \hat{N} \leqq i$$

Let's take a set I of control system queries. We assume that for every query correspond a finite nonempty set $\hat{N}_i \in \hat{N}$, such a way that

$$\bigcup_{i \in I} \hat{N}_i = \hat{N}$$

and for all $i, j \in I$, $i \neq j$ $\hat{N}_i \cap \hat{N}_j = \emptyset$

Instead of speaking about the partition $\{\hat{N}_i\}_{i \in I}$ we may consider the equivalence relation R_I $\hat{N} \times \hat{N}$ on $\hat{N}$ that the family of its equivalence classes is indexed by the set I and

$$\hat{N}/R_I = \{\hat{N}_i\}_{i \in I}$$

<u>*D E F I N I T I O N 3*</u>

A system of control system sets of data retrieval is a sixtuple

$$S_{CSSDR} = \langle D, \hat{N}, \cup, \cap, R_I, U \qquad \text{(cf.[Raś 1978])}$$

where:

D - data set

$< \hat{N}, \cup, \cap >$ - lattice of names of data subsets

R_I - an equivalence relation in and I is set of control systems queries

$U: \hat{N} \to 2^D$ - monotonic function satisfying the following conditions:

$$(\forall \hat{N}, \hat{N}' \in \hat{N})(\hat{N} \leq \hat{N}') = U(\hat{N}) \subseteq U(\hat{N})$$

$$(\forall d \in D)\ (\exists \hat{N} \in \hat{N})(d \in U(\hat{N}))$$

D E F I N I T I O N 4

Let S_{CSSDR} be a system of control system sets of data retrieval.
Let $D' \subseteq D$.

a) D' is said to be describe within S_{CSSDR} iff there is $\hat{N} \in \hat{\mathbf{N}}$ such that $U(\hat{N})=D'$.

b) $\mathbf{B}(S_{CSSDR})$ is the family of allsubsets of D descriable within S_{CSSDR}

D E F I N I T I O N 5

A system is selective iff for all $d \in D$, $U(\hat{N}_d)=\{d\}$

Theorem 2 (cf. [Lipski and Marek 1975])

A system S_{CSSDR} is selective iff $\mathbf{B}(S_{CSSDR}) = 2^D$ (recall that we consider only the case when I (set of control system queries), and consequently D, are finite).

P r o o f:

If $\mathbf{B}(S_{CSSDR}) = 2^D$ then obviously S_{CSSDR} is selective. If S_{CSSDR} is selective then all $d \in D$ have different descriptions, hence D is finite. For any $D' \subseteq D$ we have then

$$D' = U(\sum_{d \in D} \hat{N}_d),$$

i.e. D' is descriable.

5. A DATA BASE MODEL.

D E F I N I T I O N 6

Data base model of productive object is a pair

$$DB = \langle S_{DR} , S_{CSSDR} \rangle$$

where:

S_{DR} - system of dara retrieval

S_{CSSDR} - system of control system sets of data retrieval

The following properties of data base model presented above can be distinguished:

1. Model is based on a definition of relational schema of data as proposed by Codd [1971].
2. Data are the object of widely understood measurement.
3. For the presented model i.s.r, system theory can be applied.
4. Identification of data base follows the identification of information system for productive object.
5. Presented data base model can be used in the construction of distributed data base model.

Remark

Presented data base model is a proposal resulting from necessity for complex analysis of problems connected with its modelling. It concerns primarly the contents of data base.

6. DISTRIBUTED DATA BASE MODEL AND ITS PROPERTIES

DEFINITION 7

System of storage and retrieval of documents (cf. [Marek and Pawlak 1976]) is quadruple

$$I = \langle X, B, R_J, V \rangle$$

where:
- X – set of documents
- B – set of descriptors
- R_J – equivalence relation on B of finite index
- V – maps B into 2^X ($V: B \to 2^X$) and satisfied the following two conditions:
  1) if $aR_j b$ $a \neq b$, then $V(a) \cap V(b) = \emptyset$
  2) $\cup\{V(b) : bR_J a\} = X$ for each $b \in B$.

DEFINITION 8

Let $S_{CSSDR} = \langle D, \hat{M}, \cup, \cap, R_I, U \rangle$ be system of control system sets of data retrieval and I be system of storage and retrieval of documents. System I coverage S_{CSSDR} IFF

1) there exists an one-to-one function Ψ such that $\Psi: J \to I$
2) there exists a function

$$\phi : \bigcup_{j \in J} \hat{N}_{\Psi(j)} \to B \quad \text{such that}$$

$$V(b) = \cup\{U(\hat{N}) : \hat{N} \in \phi^{-1*}\{b\}\} \qquad \text{and}$$

$$\hat{N} \in \hat{N}_{\Psi(j)} \Rightarrow \phi(\hat{N}) \in B_j$$

The intuition which is connected with this following: I is a "presystem" classifying the names of data subsets in according to some documents from set of documents X.

DEFINITION 9

Let S_{CSSDR} be system of control system sets of data retrieval and I be system of storage and retrieval of documents. We say thet strongly covers S_{CSSDR} iff covers S_{CSSDR} and

$$1) \quad (\forall b)_B \quad \mathrm{card}(\phi^{-\hat{I}} * \{b\}) \geqq 2$$

and

$$2) \quad \forall i_{\ I} \quad \mathrm{card}(\hat{N}_i) \leqq 2)$$

Let K be set of indexes of existing individual (local) systems of control system sets of data retrieval.

DEFINITION 10

Let S_{CSSDR} be system of control system sets of data retrieval. Let $\{I_k\}_{k \in K}$ be a partition of the set I. An induced family $\{S_{CSSDR_k}\}_{k \in K}$ of individual (local) systems of control system sets of data retrieval is formed as follows:

$$S_{CSSDR_k} = \langle D_k, \hat{N}^{(k)}, \cup_k, \cap_k, R_{I_k}, U_k \rangle$$

$$1) \quad N^{(k)} = \bigcup_{i \in I_k} \hat{N}_i$$

$$2) \quad R_{I_k} = R_I \cap (\hat{N}^{(k)} \times \hat{N}^{(k)})$$

$$3) \quad \cup_k = \cup \upharpoonright \hat{N}^{(k)}; \quad \cap_k = \upharpoonright \hat{N}^{(k)}$$

$$4) \quad U_k = U \upharpoonright \hat{N}^{(k)}$$

$$5) \quad D_k = \hat{N}^{(k)} \in \hat{N}^k \quad U_k(\hat{N}^{(k)})$$

By S_{DR_k} will be understood restriction of system of data retrieval S_{DR} to set of data D_k

$$S_{DR_k} = \langle D_k, \mathbf{N}, \mathbf{N}^{(k)}, U_k, \mathbf{P}_k, \Theta_k, Q_k \rangle$$

where:

D_k - set of data

$\mathbf{N}$ - alphabet of data names

$\mathbf{N}^{(k)}$ - set of names of data set D_k

U_k - set of functions m_N (assigning attribute values domains) for data defined by relational schema from $\mathbf{N}^{(k)}$ set

$\mathbf{P}_k$ - set of all projections of data values for data set D_k

Θ_k - set of all two-argument order relations of data name attribute values for $\mathbf{N}^{(k)}$ set of data names

Q_k - set of set theory operations for $\mathbf{P}_k$ set of data values projections.

D E F I N I T I O N   11

Individual (local) data base model DB_k is a pair

$$DB_k = \langle S_{DR_k}, S_{CSSDR_k} \rangle$$

where:

S_{DR_k} - individual (local) system of data retrieval

S_{CSSDR_k} - individual (local) system of control system sets of data retrieval

DEFINITION 12

Divided data base model is a quadruple

$$DDB = \langle \{DB_k\}_{k \in K} , I , \Psi , \phi \rangle$$

where: $\{DB_k\}_{k \in K}$ - family of individual (local) data base models

I - system of storage and retrieval of documents (covering system)

Ψ , ϕ - covering functions

Administrator of divided data base model will be named triple

$$\bar{A} = \langle I, \Psi, \phi \rangle .$$

Final remark

Presented divided data base modes is a proposal resulting from necessity for analysis of very real problems for the geographically distributed organization of large data bases for an enterprise.

7. LITERATURE

1. Champine, G.A.: "Six approaches to distributed data bases". Datamation 5(1977) pp. 69-72.

2. Codd, E.F.: "A relational model of data for large shared data banks". Comm. ACM, Vol.13, No.6, June 1970, pp.377-387.

3. Codd, E.F.: " Further normalization of the data base relational model", Couran Comp. Sc. Symp. 6: Data base systems, Prentice-Hall, N.J., May 1971, pp. 65-98.

4. Kerschberg, L. - Klug, A. - Tsichritzis, D.: "A taxonomy of data models"; System for Large Data Bases; Lockemann, P.C.-Neuhold,E.J. (eds.), Norht-Holland Publishing Company, 1976.

5. Kobayashi, I.: "Information and information processing structure"; Information Systems 1(1975), pp.39-49.

6. Koczkodaj, W.W.: "A relational model of data and its connections with the i.s.r. systems"; ICS PAS Reports 306. Warsaw 1978.

7. Kubit, J.: "On a certain information system model"; 5th Symposium of algorithms-ALGORITHMS'79, April 1979, Strebske Pleso, pp.309-315.

8. Kuratowski, K. - Mostowski, A.: "Set theory" ; North-Holland Publishing Company, Polish Scientific Publishers, Warsaw 1976.

9. Lipski, W. - Marek, W.: "On information storage and retrieval systems"; CC PAS Report 200, Warsaw 1975.

10. Marek, W. - Pawlak, Z.: "Information storage and retrieval systems - Mathematical foundations"; Theoretival Computer Science 2(1976), pp. 331-354.

11. Marek, W. - Rode - Babczenko, I.: "A decomposition of informational systems"; CC PAS Reports 212, Warsaw 1975.

12. Neuhold, E.J.: "Formal properties of data bases"; University of Stuttgart, Mathematical Centre Tracts 63, 1974, pp.121-177.

13. Neuhold, E.J. - Biller, H.: "POREL: A distributed data base on an inhomogeneous computer network"; Third International Conference on Very Large Data Bases; Tokyo, Japan, October 1977, pp. 380-389.

14. Raś, Z.: "Algebraic foundations of information storage and retrieval system II"; (in Polish), ICP PAS Reports 339, Warsaw 1978.

ÖSSZEFOGLALÁS

Egy osztott adatbázis modell

József Kubit

Osztott adatbázisok egy formális modelljét mutatjuk be, a felbontásaival kapcsolatos alapvető jelöléseket definiáljuk. Lokális adatbázis modellek egy családját és az osztott adatbázis modell adminisztrátorát értjük osztott adatbázis modellen. A modell a relációs adatséma egy definicióján alapul.

Об одной модели распределенных баз данных

Йожеф Кубит

В данной работе представляется некоторая формальная модель распределенных баз данных, и предлагаются определения основных понятий связанных с ее декомпозицией. Модель распределенной базы данных состоит из администратора и семейства локальных моделей. Предложенная модель основана на некотором определении реляционной схемы данных.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények 24/1980.**

ИНФОРМАЦИОННЫЕ СИСТЕМЫ
SIS77 и SIS79/GENERA

Крамли А., Руда М.

1. Введение

В настоящей работе описывается две статистических системы /SIS77, SIS79/GENERA/ и дается краткая сводка о редактирующей системе GENERA (|2|, |3|) которая была использована при построении SIS79. В связи с этими статистическими системами возникли разные задачи оптимизации, одна из которых подробно исследуется в четвертой главе работы.

Основной выработки этих систем служило исследование больничного мобидитизма в Венгрии.

2. Исследования больничного морбидитизма в Венгрии

Сбор больничных данных в Венгрии начался уже в семидесятые годы прошлого века. Более подробный анализ больничных больных начинается только в начале 1950-х годов. Унифицированные сведения о статистических данных больниц восходят к 1953 году, и на базе этих данных между 1955 и 1965 гг. ежегодно совершался статистический сбор. В это время данные обрабатывались вручную и так конечно только самые простые статистические факторы были рассчитаны. Значительным шагом является сбор данных в 1967 году, когда - хотя применение машин Голлерит позволило только медленную обработку - на более сложные вопросы могли отвечать.

После первых вышеупомянутых шагов совершили первую обработку данных с помощью вычислительной машины (|1|, |4|). В 1972-73 годах через 14 месяцев зафиксированными данными работали. Проблемой выдвинулись несколько интересных вычислительных и математических вопросов, не только обработка данных большого масштаба /70-80 миллионов символов/ явилась сложной, но изготавливаемые статистические таблицы составили сложную систему. Кроме сложности, количество данных очень увеличилось.

Комплексные сведения большого количества данных несомненно необходимо, потому что народная информация есть не только в области, но и разделяется на разные части по специальности.

3. Информационная система SIS77

Из-за большого количества и сложности анализа данных больничных заболеваний, несколько зафиксированными статистическими таблицами информационных требований не удовлетворяются.

Это уже стало ясным при анализе данных 1972-1973 годов, где только зафиксированные таблицы находились. Проблема решилась упругой системой запросов |5|.

Быстрое изменение потребностей анализа морбидитизма требует разработки обобщенной, упругой системы запроса. Развитием ухода за больными и администрации больниц и изменяется форма системы представляющей данные и содержадей данные. Принимая во внимание и внешние факторы, как например, частое изменение международной классификации кодов болезней, или изменение методов, статистики народнонаселения. С изменением данных изменяется и форма и содержание информационных потребностей. Изменения организациий использующих информацию /санитарные власти, исследовательские институты/ и модифицируют потребности. Самым статистическим анализом возникаются новые информационные потребности, потому что вскрытием новых связей анализов требуются.

С 1974 года в Венгрии беспрерывно анализируются данные больничного морбидитизма. Данные больниц ежегодно обрабатываются. Беспрерывная обработка данных требовала статистического пакета программ. Разработка этого пакета на базе опытов обработки в 1972-73 годах поручили - Синистерством здравоохранения - нашему отделу. Так разрабатывалась информационная система SIS77 |8|.

Система SIS77 состоит из самостоятельных программ. Помимо изготовления статистических таблиц и проверки, кодирование и сортировка данных исполняются. Под сортировкой подразумевается оценка сложных логических функций и функций целого значения. Система дает возможность редактирования файлов. Задачи такого типа являются важными при многократном уходе за больными и анализе диагнозисов разных форм.

Программы системы являются процедурами обобщенной цели с параметрами, но обработка данных осуществляется программами генерированных вышеупомянутыми программами. Специальные программы соответствующие данной задаче при каждом прогоне снова и снова генерируются. Таким способом работа системы является эффективной. Из-за обработки сложной системы данных большого масштаба эффективности уделили большое внимание. По нашему мнению при развитии вычислительных средств интенсивные методы имеют более важную роль чем экстенсивные. Эта цель достигается осуществлением таких программных средств, с помощью которых ускоряется и облегчается работа программистов и дают возможность более экономично решить задачи. Потребность увеличения эффективности способствовала дальнейшему развитию системы SIS77. Это стремление оправдывается и его применением в рамках СЭВ-а |6|. Результатом нашей работы является информационная система SIS79/GENERA.

Перед описанием системы SIS79/GENERA мы поставим задачу об оптимизации тесно связанную с применением системы SIS77 и SIS79, и дам ее решение в одном частном случае.

4. Оптимизация разбиения систем данных

Рассмотрим множество данных состоящее из рекордов фиксированных структур и длины. Большинство задач обработки данных требует только доступа к определенным подрекордам, например, если готовится таблица сопряженности признаков используя два элементарных данных. Расходы /машинноевремя, емкость памяти/ обработки при эксплуатации системы SIS77 были сокращены с помощью разбиения файлов.

Перед формулировкой общей задачи, мы рассмотрим частный случай двухмерных таблиц.

Предположим, что основной рекорд разделен на n попарно непересекающихся полей данных имеющих длины $p_1,\dots,p_n$. Без потери общности можно предполагать, что $\sum_{i=1}^{n} p_i=1$. Далее предположим, что требуются только одномерные и двухмерные таблицы, и спаривания данных имеют одинаковые вероятности /спаривание данного с самим собой дает одновременный случай/. Если мы разделим систему данных на такие файлы, рекорды которых содержат только одну из возможных пар, то тогда чмсло файлов, нужных для изготовления любой таблицы равно $\binom{n}{2}$. Если готовится таблица относящаяся к данным из полей i и j $/i\neq j/$, то тогда расходы пропорциональны с p_i+p_j , а если оба данных принадлежат к полю i , то тогда расходы пропорциональны с выражением

$$p_i + \min_{j\neq i} p_j \ .$$

Учитывая равномерное распределение требуемых таблиц средние расходы пропорциональны с выражением.

$$/*/\quad \begin{aligned} &\sum_{i\neq j} p_i p_j (p_i+p_j) + \sum_i p_i^2 (p_i + \min_{j\neq i} p_j) = \\ &2\sum_{i=1}^{n} p_i^2 - \sum_{i=1}^{n} p_i^3 + \sum_{i=1}^{n} p_i (\min_{j\neq i} p_j) \ . \end{aligned}$$

Используя неравенство Йенсена и элементарные методы дифференциального исчисления легко проверить, что в случае $n \geq 4$ выражение /ж/ достигает минимум для прспределения длин p_i вида

$$p_1 = \dots = p_{n-1} = p\ , \quad p_{n-1} = q\frac{1-2q}{8(n-2)},\ p_n = q + \frac{1-2q}{8(n-2)}\ .$$

Например, если $n = 4$, то тогда $p = 0.2713$, $q = 0.2287$, и минимум равен 0.495, который немножко меньше чем число полученное при равномерном разделении /0.5/ Наш пример показывает, что в этом простом случае оптимум достигается не для наглядно ожидаемого разделения.

Сейчас мы сформулируем более общую задачу. Пусть фиксирована грань K. Предположим, что $\{A_i\}$ является системой подрекордов основного рекорда, которая достаточна для изготовления любой таблицы сопряженности признаков и

/жж/ $$\Sigma\ p(A_i) < K$$

/$p(A_i)$: длина рекорда A_i, расходы обработки файла с подрекордами типа A_i пропорциональны с $p(A_i)$/.

Определить оптимальную систему $\{A_i\}$ удовлетворяющую условию /жж/, если $p(A_i)$ может быть любое действительное число < 1, или если $p(A_i)$ равно одному из рациональных чисел k/n, $k = 0,\dots, n$ /n фиксировано/.

Что делать если вероятности доступа к разным элементарным данным независимы , но и не одинаковы.

5. Информационная система SIS79/GENERA

Информационная система SIS79/GENERA |2|, |3| является развитым вариантом системы SIS77. С точки зрения способа использования эти две системы значительно отличаются друг от друга. Кроме этого SIS79 включает в себя новые процедуры и процедуры системы SIS77 являются более эффективными. Направления дальнейшего развития следующие:

1. В первую очередь в помощью ускорения работы процедуры эффективность подсистемы SIS77 увеличивается. В этой области результаты достигнуты в связи с вводными/выводными операциями и сложными функциями с многими переменными.

2. Редакционная система системы SIS79/GENERA помогает эффективности работы. Эта система, сравнивая с системой SIS77, оказывает значительное значение на уровнях системных программистов и пользователей. Система GENERA прекращает сравнительно жесткую систему самостоятельных, но тесно связанных программ системы SIS77, и дает возможность свободно решить задачи. Организующая система GENERA, с одной стороны, облегчает изготовление процедур для генерирования программы /и так в значительной мере помогает работе системных программистов/, а с другой стороны, работа изготовленных процедур организуется /к программе написанной на базисном языке специальные инструкции стыкуются, распознающие подпрограммы которых связаны с GENERA. Распознающие программы входят в состав препроцессора/ SIS79 составит в себе подпрограммы вышеупомянутого типа помогающих описанию статистических задач. Результатом работы GENERA является прграмма на базисном языке /например: ФОРТРАН/. В систему GENERA включается процессор /подпрограммы для распознования инструкции/ не только статистического, а какого-либо типа, который стыкуется с базисным языком. Таким образом система GENERA является средством широкого пользования.

3. Система SIS77 представляет собой закрытую систему. Для стыковки с другой системой - например, математико-статистическим пакетом программ, процедурами для построения таблиц - целесообразно осуществить интерфейсы. С помощью таких интерфейсов имеется возможность стыковаться с отличными пакетами программ как BMDP или SPSS.

4. Система SIS77 осуществлена на ЭВМ HONEYWELL 66/60, и анализы больничного морбидитизма совершились на этой же ЭВМ. Более большая упругость системы SIS79/GENERA позволяет ее пользование в более широкой области, по этому варианту работающие на ЭВМ ЕС и IBM и разрабатываются. Это дает возможность использования системы GENERA в других системах.

Коротко изложим самые важные принципы организации системы SIS79/GENERA.

Сперва разрешите сказать несколько слов о нашем методе генерирования программ. Повторяющееся генерирование обрабатывающих программ дает, в первую очередь, возможность уменьшить требования к ЭВМ /время и память/. Важным фактором является простое построение яркая структура генерерования программ.

При уходе за системой этот факт является очень важным, потому что выгодно раскрываются и исправляются ошибки. В конце концов генерирование программы более упруго соответствуют различным потребностям чем системы, работающие в зафиксированной форме, и так в занчительной мере помогают работе программистов.

Другим нашим основным принципом является обеспечение модулярности системы. Уже система SIS77 содержит в себе программы иерархической организации, с разными возможностями пользования. Модулярность и упругость метода генерирования программ оказали влияние на успешное использование системы. Анализ больничного морбидитизма совершен не только в народных условиях, а успешно осуществили его в рамках медицинского отдела СЭВ-а. Этот анализ базировался на данных больных людей 1 миллионной популяции во время одного месяца /май 1979 г./. Этот анализ осуществился в рамках темы СЭВ-а: "Сбор, анализ и использование данных общего морбидитизма населения, разработка унифицированных методов для стран-членов СЭВ".

Препроцессорная структура SIS79 позволяет разделение задач на элементы. Инструкции SIS79 встроенные в хост язык ФОРТРАН обеспечивают описание любой комбинации задач.

Нашей основной целью являлось минимальным количеством подсистем описать операции статистической обработки и подготовки данных. Система SIS79/GENERA включает в себя следующие подсистемы:

1. Вычисление /контроль, кодирование/ сложных функций со многими переменными целого значения, определенных только с помощью таблиц значения. Вычисление сложных функций со многими переменными целого значения определенных только с помощью таблиц значения часто требует хранения и анализа сложных таблиц значения большого масштаба. В таких случаях является целесообразным стремиться к более лучшему использованию памяти, к уменьшению времени прогона, специально тогда, когда анализ много раз осуществляется на базе данных большого масштаба. Методом репрезентации функций в наших системах одновременно обеспечиваются уменьшение времени прогона и емкости памяти. Следует отметить, что удовлетворение этих двух противоречивых условий в общем является очень сложным вопросом.

2. Специальные операции ввода/вывода

   При обработке данных значительная часть емкости ЭВМ очень часто служит для хранения, движения и преобразования данных. Быстрые процедуры преобразования системы SIS79 в 5-10 раз быстрее чем соответствующие вводные/выводные операции языка ФОРТРАН. Особенно тщательно занимались со сжатием хранения. В системе SIS79 имеется возможность сжатого бинарного представления данных, с помощью которого затребованное количество памяти значительно уменьшается. С использованием положительных целых чисел в кодировании в медицинских и других задачах обработки данных позволяет быстрое преобразование и сжатое хранение.

3. Вычисление сложных логических выражений

   При обработке больничных данных для изготовления статистики часто выделяется часть популяции /например: сельскохозяйственное население одного комитата/. Выбор части популяции осуществляется по простыми или сложными логическими условиями. описание простых логических языков целесообразно решать на традиционных языках программирования /например: в системе SIS79 язык ФОРТРАН/. При более сложных выражениях, специально когда повторно много раз вычисляются, целесообразным является выработка более эффективного средства. В наших статистических системах /SIS77, SIS79/GENERA/ оценка любых логических выражений нормальной формы эффективно решается |7| .

4. При изготовлении статистических таблиц стремились достигнуть максимальную эффективность. Это стремление оправдывается фактом, что при анализе больничного морбидитизма количество данных несколько сотен тысяч и количество группы таблиц ежегодно 100-200. Наша цель осуществляется с использованием табличного файла /преобразованный файл/. В табличных файлах примером является совокупность людей находящихся в одной группе возрата, той же самой болезни, профессии и места жительства. Статистические таблицы получаются сокращением соответствующих типов из табличного файла.

   Техника табличного файла позволяет получить статистические таблицы - независиом от количества данных - через несколько машинных секунд /на ЭВМ HONEYWELL 66/60/.

В заключении следует отметить, что информационная система SIS77 является эффективным вычислительным средством в анализе венгерских больничных морбидитизмов и в анализе высокого давления у молодежи в рамках СЭВ-а. Эти благоприятные опыты и возможности дальнейшего развития оказали влияние на осуществление системы SIS79/GENERA.

ЛИТЕРАТУРА

[1] Csukás M., Greff L., Krámli A., Ruda M., An approach to the hospital morbidity data system development in Hungary, Symposium on Medical Data Processing, Toulouse, 1975.

[2] Kerékfy P., Система SIS79/GENERA, I.заседание международной рабочей группы № 16 /РГ 16/ "Обработка биомедицинских данных с помощью ЭВМ" и научноучебная конференция, Jadwisin, 1980.

[3] Kerékfy P., GENERA – A Program Generator System, Fifth European Meeting on Cybernetics and System Research, Vienna, 1980.

[4] Krámli A., Ruda M., The computer realisation and first experiences of the hospital morbidity study, WHO Статистический Передвижной Семинар, Budapest, 1974.

[5] Krámli A., Ruda M., Справочно-информационная система запросов больничного мобидитизма, Структура и организация пакетов программ, Международная конференция, Тбилиси, 1976.

[6] Krámli A., Ruda M., Csukás M., Galambos M., Large samle size statistical system for HWB, Second Interbational Symposium on Data Analysis and Informatics, Versailles, 1979.

[7] Ratkó I., On optimization problems of logical expressions in programing languages, Конференция по математической логике в теории программирования, Salgótarján /Венгрия/, 1979.

[8] Ruda M., Statistical Information System with Health Service Application, 4-th Winterschool of Visegrád on the Theory of Operating System, Szentenre, 1978.

S u m m a r y

SIS77 and SIS79/GENERA information systems

Krámli A., Ruda M.

In this paper there are presented two statistical information systems (SIS77, SIS79) and a program redacting system GENERA that was used for building up the system SIS79. After summarizing briefly the Hungarian Hospital Morbidity Stuy, which was the first task solved by SIS77 we state and solve some optimization problems connected with the applications of these systems.

Ö s s z e f o g l a l ó

A SIS77 és a SIS79/GENERA információs rendszer

Krámli András, Ruda Mihály

A dolgozat két információs rendszert mutat be (SIS77, SIS79), és röviden tárgyalja a GENERA programkészitő rendszert is, amely a SIS79 felépitésének alapjául szolgál. A statisztikai rendszerek alkalmazásával kapcsolatban számos optimalizálási probléma merül fel. Ezek közül tárgyalunk egyet (adatrendszerek optimális felbontásának kérdéseit) dolgozatunk negyedik fejezetében. Egy rövid áttekintést adunk a két rendszer fejlesztésének alapjául szolgáló országos morbiditási vizsgálatokról is.

MTA Számítástechnikai és Automatizálási Kutató Intézete, **Közlemények 24/1980.**

ОБОБЩЕНИЕ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ В РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ

Я. Деметрович, Дь. Дьепеши

§ 0. Введение и основные определения

Реляционная модель данных, введенная Е.Ф. Коддом [7] , является одним из самых многообразных средств обработки данных. Эта модель выдвигает на первый план не машинную эффективность, а наглядное описание данных с точки зрения пользователя. Хранящиеся данные наглядно изображаются в виде матриц. Строки матриц соответствуют записям, а столбцы - свойствам или атрибутам, характерным для данного типа записей.

Дадим теперь строгое определение. Пусть Ω - конечное, непустое множество - <u>множество атрибутов</u>. Для каждого $a\in\Omega$ обозначим через $D(a)$ непустое множество, являющееся <u>множеством значений атрибута a</u>. Подмножества декартового произведения $\underset{a\in\Omega}{X}D(a)$ называется <u>реляциями над множеством Ω</u>. Таким образом реляция R над Ω состоит из таких функций $g\colon \Omega\to \underset{a\in\Omega}{U}D(a)$, для которых $g(a)\in D(a)$, если $a\in\Omega$.

Функции, составляющие реляцию R, мы называем <u>строками R.</u>

Таким образом реляцию R под множеством $\Omega = \{a_1,\dots, a_n\}$ можно представить в виде двухмерной таблицы:

	a_1	a_2	...	a_n
g	$g(a_1)$	$g(a_2)$	...	$g(a_n)$
⋮				
h	$h(a_1)$	$h(a_2)$	...	$h(a_n)$
⋮				

Понятие функциональной зависимости Е.Ф. Кодд [7] ввел следующим образом.

Пусть R-реляция над Ω и пусть $A, B \subseteq \Omega$. Будем говорить, что <u>B функционально зависит от A в R</u> /обозначается: $A \underset{R}{\overset{f}{\rightarrow}} B$/, если для любых $g, h \in R$

(1) $(\forall a \in A)(g(a) = h(a)) \Rightarrow (\forall b \in B)(g(b) = h(b))$

(1) означает, что

(1′) если две строки равны <u>на всех элементах A</u>, то они равны и <u>на всех элементах B</u>.

На основе формального определения ((1)) и значения функциональной зависимости можно естественным образом определить еще три вида зависимостей, которые характеризуют реляции.

Заметим, что в (1) присутствуют два квантора $\forall$. Заменяя их всеми возможными способами на кванторы $\exists$ и $\forall$, мы получим формулы, имеющие смысл:

(2) $(\exists a \in A)(g(a) = h(a)) \Rightarrow (\exists b \in B)(g(b) = h(b))$

(3) $(\exists a \in A)(g(a) = h(a)) \Rightarrow (\forall b \in B)(g(b) = h(b))$

(4) $(\forall a \in A)(g(a) = h(a)) \Rightarrow (\exists b \in B)(g(b) = h(b))$

(2), (3) и (4) означают соответственно следующее:

(2') если две строки равны на некотором элементе A, то они равны и на некотором элементе B.

(3') если две строки равны на некотором элементе A, то они равны на всех элементах B.

(4') если две строки равны на всех элементах A, то они равны на некотором элементе B.

По образцу определения функциональной зависимости на основе (2), (3) и (4) дадим определение дуальной зависимости ($A \overset{d}{\underset{R}{\to}} B$) и сильной зависимости ($A \overset{s}{\underset{R}{\to}} B$) и слабой зависимости ($A \overset{w}{\underset{R}{\to}} B$).

Если R - реляция над Ω, то пусть

$$\mathbf{F}_R = \{(A, B): A \overset{f}{\underset{R}{\to}} B\}$$

$$\mathbf{D}_R = \{(A, B): A \overset{d}{\underset{R}{\to}} B\}$$

$$\mathbf{S}_R = \{(A, B): A \overset{s}{\underset{R}{\to}} B\}$$

$$\mathbf{W}_R = \{(A, B): A \overset{w}{\underset{R}{\to}} B\}$$

Если $\mathbf{Y} \in \{\mathbf{F}, \mathbf{D}, \mathbf{S}, \mathbf{W}\}$, $y \in \{f, d, s, w\}$ и y соответствует $\mathbf{Y}$ /например, если $\mathbf{Y} = \mathbf{D}$, то $y = d$/, то множество вида $\mathbf{Y}_R$ мы называем полными y-семействами.

Естественна потребность определить для полных y-семейств наиболее общие свойства - т.е. выполяющиеся для $\mathbf{Y}_R$ при любых реляциях R - таким образом, чтобы эти свойства описали все полные y-семейства. Другими словами, если некоторое $\mathbf{Y} \subseteq P(\Omega) \times P(\Omega)$ обладает этими свойствами, то должна существовать такая реляция R над Ω, что $\mathbf{Y} = \mathbf{Y}_R$. Короче говоря, мы хотели аксиоматизировать y-семейства для любого $y \in \{f, d, s, w\}$.

Математическую структуру функциональных зависимостей впервые изучал В.В. Армстронг [1] . Он аксимаьизировал полные f-семейства /см. § 1, φ -аксиомы/ и доказал, что полное f-семейство определяется системой своих максимальных зависимостей /таких зависимостей, зависимая часть которых максимальна/, и даже системой своих максимальных правых частей /зависимая часть максимальной зависимости/.

Г. Цедли в [8] , аналогичным способом аксиоматизировал полные d- и s-семейства. Аксиоматизация полных w-семейств тем же путем не удалась.

В этой работе мы дадим новые аксиомы для характеристики y-семейств для любого $y \in \{f,d,s,w\}$ /в § 1 F-, D-, S- и W-аксиомы, а в § 2 F'-, D'-, S'- и W'- аксиомы/. Эти аксиомы выделяют комбинаторную структуру y-семейств, в то время как системы аксиом Армстронг и Цедли подчеркивают их логическую структуру.

Нам кажется, что комбинаторный подход более пригоден для рассмотрения возникающих здесь проблем. Кроме того, таким путем удалось аксиоматизировать полные w-семейства.

В § 1 мы приведем новые аксиомы /F-, D-, S- аксиомы/ для полных f-, d- и s-семейств, а также докажем, что они эквивалентны аксиомам, которые ввелись [1, 5, 8] .

В § 2 мы введем понятие равенственного множества реляции, охарактеризуем равенственные множества комбинаторным путем /теорема 2.1./ и, основываясь на этом понятии, введем F'-, D'-, S'- и W'- аксиомы. Будет доказано, что F'-, D'-, и S'-аксиомы эквивалентны, соответственно, φ -аксиомам ϑ и γ -аксиомам. В теореме 2.2. мы укажем на одно существенное отличие полных w-семейств от остальных полных семейств.

В теореме 2.3. мы покажем, что F'-, D'-, S'- и W'- аксиомы действительно описывают полные f-, d-, s- и w-семейства.

В дальнейшем часто употребляем следующие обозначения:

Если в область определения строки g $(g \in R)$ $A \subseteq \Omega$, то обозначим через $g\restriction_A$ те строки, область определения которой совпадает с A и для которой $(\forall a \in A)\ g\restriction_A(a) = g(a)$.

В заключении сделаем несколько практических замечаний. Функциональные, дуальные, сильные и слабые зависимости являются такими ограничениями, знание которых дает возможность охарактеризовать реляции с помощью ограничений, налагаемых на некоторое подмножество полного множества столбцов.

Проектировщики реляционных баз данных знакомы с некоторыми зависимостями, имеющими место в базах данных. Обычно эти зависимости не образуют полного семейства, то есть в базе данных наверняка имеют место и другие зависимости. Их определение и экономичное хранение является одной из основных проблем.

Отметим также, что теория зависимостей тесно связана с одной исключительно трудной комбинаторной проблемой: с определением мощности так называемых Δ - систем.

§ 1. Старые и новые аксиомных разных зависимостей.

В этом параграфе, мы опишем известные аксиомы функциональных [1] , двойственных и сильных систем зависимостей [8] , а затем приведем новые аксиомы /F-,D-,S - аксиома/ эквивалентные им. Дальнейший анализ новых аксиом приведет к абстактному описанию слабых систем зависимостей.

На основе функциональных зависимостей /φ - аксиом/ казалось, что проблемы теории разных - функциональных, двойственных, сильных и слабых - зависимостей имеют логический характер [2,3,11] и по нашему мнению их легче рассматривать на основе **F**-аксиомы.

Сходство **F-**, **D-**, **S-** аксиом легко описывается, поэтому нетрудно перенести на случай функциональных, двойственных зависимостей понятия и результаты, изветсные для функциональных зависимостей. Согласно **F'- D'- S'- W'** аксиомам /§ 2/ то же самое можно сделать и для слабых зависимостей.

Начнем с определения функциональных, двойственных, сильных и слабых зависимостей.

Определение 1.1. Пусть даны подмножества A, B ($A, B \subseteq \Omega$) и реляция R над множеством Ω. Мы говорим, что множество B

(i) функционально /обозначается: $A \overset{f}{\underset{R}{\to}} B$ /

(ii) двойственно /обозначается: $A \overset{d}{\underset{R}{\to}} B$ /

(iii) сильно /обозначается: $A \overset{s}{\underset{R}{\to}} B$ /

(iv) слабо /обозначается: $A \overset{w}{\underset{R}{\to}} B$ /

зависит от множества A в реляции R, если соответственно выполняется следующее условие:

(i) $(\forall g, h \in R)\ (g\restriction_A = h\restriction_A \Rightarrow g\restriction_B = h\restriction_B)$

(ii) $(\forall g, h \in R)\ ((\exists a \in A)\ (g(a) = h(a)) \Rightarrow (\exists b \in B)\ (g(b) = h(b)))$

(iii) $(\forall g, h \in R)\ ((\exists a \in A)\ (g(a) = h(a)) \Rightarrow g\restriction_B = h\restriction_B)$

(iv) $(\forall g, h \in R)\ (g\restriction_A = h\restriction_A \Rightarrow (\exists b \in B)\ (g(b) = h(b)))$.

Обозначим зависимости типа $A \underset{R}{\overset{f}{\to}} B$, $A \underset{R}{\overset{d}{\to}} B$, $A \underset{R}{\overset{s}{\to}} B$, $A \underset{R}{\overset{w}{\to}} B$ соответственно через

$$\mathbf{F}_R = \{(A, B) : A \underset{R}{\overset{f}{\to}} B\}, \quad \mathbf{D}_R = \{(A, B) : A \underset{R}{\overset{d}{\to}} B\},$$

$$\mathbf{S}_R = \{(A, B) : A \underset{R}{\overset{s}{\to}} B\}, \quad \mathbf{W}_R = \{(A, B) : A \underset{R}{\overset{w}{\to}} B\}$$

и назовем каждое из этих множеств полным семейством данного типа относительно реляции $\mathbf{R}$.

Для простоты введем следующие обозначения:

$$\mathbf{Y}_R \in \{\mathbf{F}_R, \mathbf{D}_R, \mathbf{S}_R, \mathbf{W}_R\}.$$

Тогда

$$\mathbf{Y}_R = \{(A, B) : A \underset{R}{\overset{y}{\to}} B\}, \text{ где } y \in \{f, d, s, w\}.$$

Семейство $\mathbf{Y}_R$ назовем полным семейством типа y.

Разные зависимости до сих пор изучались во многих работах [4,10,12,13]. Основной целью этих работ являлась аксиоматизация полных семейств этих зависимостей [1,6] и изучение их свойств [3,4,5,9]. Аксиоматизация функциональных зависимостей велась в двух основных направлениях: отыскание протых аксиом, с которыми легко работать [1], а также разработка минимальных систем аксиом [6].

Армстронг в работе [1] аксиоматизировал полные семейства функционального типа и в работе [8] были даны некоторые системы аксиом для полных семейств двойственного и сильного типа. В дальнейшем мы приведем системы аксиом для полных семейств функционального, двойственного, слабого и сильного типа, а также покажем, что эти новые аксиомы эквивалентны старым. Для полноты опишем старые аксиомы.

Пусть дано семейство подмножества $\mathbf{Y}$, где $\mathbf{Y} \subseteq P(\Omega) \times (\Omega)$ и $P(\Omega)$ семейство всех подмножеств над множеством Ω.

Будем говорить, что семейство $\mathbf{Y}$ удовлетворяет φ-аксиомам, тогда и только тогда, когда для каждого из множеств A, B, C, D ($A,B,C,D \subseteq \Omega$) выполняются следующие условия:

(F1) $(A,A)\in\mathbf{Y}$

(F2) $(A,B)\in\mathbf{Y},(B,C)\in\mathbf{Y} \Rightarrow (A,C)\in\mathbf{Y}$

(F3) $(A,B)\in\mathbf{Y},\ C \supseteq A,\ D \subseteq B \Rightarrow (C,D)\in\mathbf{Y}$

(F4) $(A,B)\in\mathbf{Y},\ (C,D)\in\mathbf{Y} \Rightarrow (A \cup C,\ B \cup D)\in\mathbf{Y}$

Будем говорить, что семейство $\mathbf{Y}$ удовлетворяет ϑ-аксиомам тогда и только тогда, когда для каждого из множеств A, B, C, D выполняюся следующие условия:

(D1) $(A,A)\in\mathbf{Y}$.

(D2) $(A,B)\in\mathbf{Y},\ (B,C)\in\mathbf{Y} \Rightarrow (A,C)\in\mathbf{Y}$

(D3) $(A,B)\in\mathbf{Y},\ C \subseteq A,\ B \subseteq D \Rightarrow (C,D)\in\mathbf{Y}$

(D4) $(A,B)\in\mathbf{Y},\ (C,D)\in\mathbf{Y} \Rightarrow (A \cup C,\ B \cup D)\in\mathbf{Y}$

(D5) $(A,\emptyset)\in\mathbf{Y} \Rightarrow A = \emptyset$

Будем говорить, что семейство $\mathbf{Y}$ удовлетворяет γ-аксиомам, тогда и только тогда, когда для каждого из множеств A,B,C,D и для каждого $a\in\Omega$ выполняются следующие условия:

(S1) $(\{a\},\{a\})\in\mathbf{Y}$

(S2) $(A,B)\in\mathbf{Y},\ (B,C)\in\mathbf{Y},\ B \neq \emptyset \Rightarrow (A,C)\in\mathbf{Y}$

(S3) $(A,B)\in\mathbf{Y},\ C \subseteq A,\ D \subseteq B \Rightarrow (C,D)\in\mathbf{Y}$

(S4) $(A,B)\in\mathbf{Y},\ (C,D)\in\mathbf{Y} \Rightarrow (A \cap C,\ B \cup D)\in\mathbf{Y}$

(S5) $(A,B)\in\mathbf{Y},\ (C,D)\in\mathbf{Y} \Rightarrow (A \cup C,\ B \cap D)\in\mathbf{Y}$

φ- ϑ- и γ-аксиомы описывают свойства системы функциональных, двойственных и сильных зависимостей, независимые от специальной структуры реляций. Это означает, что во-первых для всякой реляции R над множеством Ω полное семейство функциональных зависимостей $\mathbf{F}_R$ удовлетворяет φ-аксиомам [1,6], полное семейство двойственных зависимостей $\mathbf{D}_R$ удовлетворяет ϑ -аксиомам [6] и полное семейство сильных зависимостей $\mathbf{S}_R$ удовлетворяет γ -аксиомам [8] ; а во-вторых, если семейство $\mathbf{F}$ удовлетворяет φ-аксиомам, то существует такая реляция R, что $\mathbf{F} = \mathbf{F}_R$ [1,10] ; если семейство $\mathbf{D}$ удовлетворяет ϑ-аксиомам, то существует такая реляция R, что $\mathbf{D} = \mathbf{D}_R$ [8] и если $\mathbf{S}$ удовлетворяет γ-аксиомам, то существует такая реляция R, что $\mathbf{S} = \mathbf{S}_R$ [8] .

В дальнейшем нам понадобится несколько технических лемм.

Лемма 1.1. Пусть семейство $\mathbf{F} \subseteq P(\Omega) \times P(\Omega)$ такое, что если $(A,B)\in\mathbf{F}$ и $B \neq \emptyset$, то $A \neq \emptyset$. Тогда семейство $\mathbf{F}$ удовлетворяет φ-аксиомам тогда и только тогда, когда семейство $\mathbf{D} = \{(B, A) : (A, B)\in \mathbf{F}\}$ удовлетворяет ϑ-аксиомам.

Доказательство. Утверждение леммы легко вытекает из φ и ϑ аксиом . Из-за аксиомы (D5) надо предполагать, что $((A, B)\in\mathbf{F},\ B \neq \emptyset \rightarrow A \neq \emptyset)$ условие выполняется. □

Замечание. Пусть семейство $\mathbf{F}$ удовлетворяет φ-аксиомам. Обозначим через $\mathbf{F}' = \mathbf{F} \setminus \{(\emptyset, B) : B \neq \emptyset\}$. Тогда семейство $\mathbf{F}'$ удовлетворяет условиям леммы 1.1. и если $A \subseteq \Omega$, $A \neq \emptyset$, тогда $(\forall B \subseteq \Omega)((A,B) \in \mathbf{F} \leftrightarrow (A,B)\in\mathbf{F}')$. Заметим еще, что если семейство $\mathbf{F}$ удовлетворяет φ-аксиомам, то к $\mathbf{F}$ естественным образом можно поставить в соответствия некоторое семейство $\mathbf{D}(\mathbf{F})$, для которого выполняются ϑ-аксиомы, таким образом, что $\mathbf{D}(\mathbf{F}) = \{(B,A) : (A,B)\in\mathbf{F}\ \&\ A = \emptyset \rightarrow B = \emptyset\}$. Естественно, что при этом "одно и то же $\mathbf{D}$ может соответствовать более чем одному семейству $\mathbf{F}$ ".

В дальнейшем вместо системы аксиом φ, ϑ и γ мы определим новые системы аксиом и дадим систему аксиом для полных семейств слабого типа, которые не содержат зависимости, где первое подмножество В пустое т.е. $(\emptyset \overset{W}{\underset{R}{\to}} B)$.

<u>F - аксиома</u>:

Семейство $\mathbf{F} \subseteq P(\Omega) \times P(\Omega)$ удовлетворяет F-аксиомам тогда и только тогда, когда для каждого $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$ существует такое множество $E (E \subseteq \Omega)$, что выполняются следующие условия:

(i) $X \subseteq E$ и $Y \not\subseteq E$

(ii) если $(A, B) \in \mathbf{F}$ и $A \subseteq E$ то $B \subseteq E$.

<u>D - аксиома</u>.

Семейство $\mathbf{D} \subseteq P(\Omega) \times P(\Omega)$ удовлетворяет D-аксиомам тогда и только тогда, когда если для каждого $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}$ существует такое множество $E (E \subseteq \Omega)$, что выполняются следующие условия:

(i) $X \cap E \neq \emptyset$ u $Y \cap E = \emptyset$

(ii) если $(A, B) \in \mathbf{D}$ u $A \cap E \neq \emptyset$, то $B \cap E \neq \emptyset$

<u>S - аксиома</u>

Семейство $\mathbf{S} \subseteq P(\Omega) \times P(\Omega)$ удовлетворяет S-аксиомам тогда и только тогда, когда если для каждого $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$ существует такое множество $E (E \subseteq \Omega)$, что выполняются следующие условия:

(i) $X \cap E \neq \emptyset$ и $Y \not\subseteq E$

(ii) если $(A, B) \in \mathbf{S}$ и $A \cap E \neq \emptyset$, то $B \subseteq E$.

W - аксиома.

Семейство $\mathbf{W} \subseteq P(\Omega) \times P(\Omega)$ удовлетворяет W-аксиомам тогда и только тогда, когда если для каждого $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{W}$ существует такое множество $E (E \subseteq \Omega)$, что выполняются следующие условия:

(i) $X \subseteq E$ и $Y \cap E = \emptyset$

(ii) если $(A, B) \in \mathbf{W}$ и $A \subseteq E$, то $B \cap E \neq \emptyset$.

Покажем, что новые аксиомы эквивалентны старым.

Теорема 1.1.

(i) Пусть дано семество $\mathbf{F}$. $\mathbf{F}$ удовлетворяет φ-аксиомам тогда и только тогда, когда $\mathbf{F}$ удовлетворяет F-аксиомам .

(ii) Пусть дано семейство $\mathbf{D}$. $\mathbf{D}$ удовлетворяет ϑ аксиомам тогда и только тогда, когда $\mathbf{D}$ удовлетворяет D-аксиомам.

(iii) Пусть дано семейство $\mathbf{S}$. $\mathbf{S}$ удовлетворяет γ-аксиомам тогда и только тогда, когда $\mathbf{S}$ удовлетворяет S-аксиомам.

Доказательство

(i) Легко проверить что, если семейство $\mathbf{F}$ удовлетворяет F-аксиомам, тогда $\mathbf{F}$ удовлетворяет φ-аксиомам тоже. Например, покажем, что если семейство $\mathbf{F}$ удовлетворяет F-аксиомам, то аксиома (F2) выполняется. Доказательство проведем от противного. Пусть $(A, B) \in \mathbf{F}$, $(B, C) \in \mathbf{F}$ и $(A, C) \notin \mathbf{F}$. Тогда исходя из того, что семейство $\mathbf{F}$ удовлетворяет F-аксиоме, существует такое множество E $(E \subseteq \Omega)$, что $A \subsetneq E$, $C \nsubseteq E$. А из того, что $(A, B) \in \mathbf{F}$ и $(B, C) \in \mathbf{F}$ следует, что

$$((A, B) \in \mathbf{F}, A \subseteq E \to B \subseteq E), (B, C) \in \mathbf{F}, B \subseteq E \to C \subseteq E).$$

А это противоречие.

Предположим, что семейство $\mathbf{F}$ удовлетворяет φ-аксиомам и пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$. Покажем, что тогда

(1) существует такое множество $E \supseteq X$, что $(E, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$ и если $E' \supset E$, то $(E', Y) \in \mathbf{F}$.

Действительно, согласно аксиомы (F1) $(\Omega, \Omega) \in \mathbf{F}$, поэтому из аксиомы (F3) следует, что $(\Omega, Y) \in \mathbf{F}$. Таким образом $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$ и $(\Omega, Y) \in \mathbf{F}$ т.е. существует такое максимальное $E \supseteq X$, для которого $(E, Y) \in \mathbf{F}$. Тем самым утверждение (1) доказано.

Пусть $\mathbf{F}$ удовлетворяет φ-аксиомам и пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus F$. Мы утверждаем, что множество E $(E \supseteq X)$ удовлетворяет F-аксиоме из-за утверждения (1). Если $Y \subseteq E$, то из аксиом (F1) и (F3) следует, что $(E, Y) \in \mathbf{F}$, в то время, как согласно утверждению (1) $(E, Y) \notin \mathbf{F}$; таким образом $Y \nsubseteq E$.

Пусть $(A, B) \in \mathbf{F}$. Предположим, что $A \subseteq E$ и $B \nsubseteq E$. Тогда $E' = E \cup B \supset E$ и значит $(E', Y) \in \mathbf{F}$. Согласно аксиоме (F1) $(E, E) \in \mathbf{F}$, поэтому из аксиомы (F4) следует, что $(E, E') = (E \cup A, E \cup B) \in \mathbf{F}$. Но тогда, согласно аксиоме (F2) $((E, E') \in \mathbf{F}, (E', Y) \in \mathbf{F} \to (E, Y) \in \mathbf{F})$, что является противоречием. Таким образом $B \subseteq E$ и тем самым доказательство (i) закончено.

(ii) Пусть $\mathbf{F} = \{(A, B) : (B, A) \in \mathbf{D}\}$. Согласно лемме 1.1. семейство $\mathbf{F}$ удовлетворяет φ-аксиомам тогда и только тогда, когда семейство $\mathbf{D}$ удовлетворяет ϑ - аксиомам. Вследствие пункта (i) достаточно доказать, что семейство $\mathbf{F}$ удовлетворяет F-аксиоме тогда и только тогда, когда семейство $\mathbf{D}$ удовлетворяет D-аксиоме. Предположим, что для семейства $\mathbf{F}$ справедлива F-аксиома. Согласно F-аксиоме, для $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$

существует такое множество E $(E \subseteq \Omega)$, что $X \subseteq E$, $Y \subsetneq E$ и если $(A, B) \in \mathbf{F}$ и $A \subseteq E$, то $B \overset{\circ}{\subsetneq} E$. В дальнейшем обозначим через $E(X, Y)$, выше определенное множество E, т.е. $E(X, Y) = E$. Тогда для $(Y, X) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}$ множестве $(\Omega \setminus E\ (X, Y))$ такое будет для которого выполяется D-аксиома. Поскольку $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$ <–> $(Y, X) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}$, то из этого следует, что $\mathbf{D}$ удовлетворяет D-аксиоме.

Аналогично доказывается, что если семейство $\mathbf{D}$ удовлетворяет D-аксиоме, то семейство $\mathbf{F}$ удовлетворяет F-аксиоме. Таким образом, пункт (ii) полностью доказан.

(iii) Предположим, что семейство $\mathbf{S}$ удовлетворяет S-аксиоме.

Нетрудно проследить, что тогда $\mathbf{S}$ удовлетворяет γ-аксиоме, аналогично тому, как это было сделано для пункта (i). Остановимся подробнее на доказательстве аксиомы (S1). Предположим, что существует такое $a (a \in \Omega)$, для которого $(\{a\}, \{a\}) \notin \mathbf{S}$. Тогда согласно S-аксиоме существует такое $E (E \subseteq \Omega)$, что $\{a\} \cap E \neq \emptyset$ и $(\Omega \setminus E) \cap \{a\} \neq \emptyset$. А это противоречит тому, что $|\{a\}| = 1$. Следовательно, $(\{a\}, \{a\}) \in \mathbf{S}$.

Предположим теперь, что семейство $\mathbf{S}$ удовлетворяет γ-аксиомам и пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$. Покажем, что тогда (2) существуют $a (a \in X)$ и множество $E (E \subseteq \Omega)$ такие, что $a \in E$, $(\{a\}, E) \in \mathbf{S}$ и если $E' \supset E$, то $(\{a\}, E') \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$.

Действительно, если для любого $a \in X$ справедливо, что $(\{a\}, Y) \in \mathbf{S}$, то последовательное применение аксиомы (S5) дает нам, что $(X, Y) \in \mathbf{S}$. Таким образом существует такое $a \in X$, что $(\{a\}, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$. Если для всякого $b (b \in Y)$ выполняется, что $(\{a\}, \{b\}) \in \mathbf{S}$, то последовательно применяя аксиомы (S4) получаем, что $(\{a\}, Y) \in \mathbf{S}$; то есть существует такое $b \in Y$, что $(\{a\}, \{b\}) \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$. Из аксиом (S1) и (S2) следует, что существует такое множество E, что $a \in E$ и $(\{a\}, E) \in S$, причем E является максимальным множеством обладающим этим свойством. Тем самым утверждение (2) доказано.

Пусть семейство S удовлетворяет γ-аксиомам и пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus S$. Мы утверждаем, что тогда семейство S удоволетворяет S-аксиоме и для множества E, определенного в утверждении (2), выполняются условия (i) и (ii) S-аксиомы.

Из $a \in X \cap E$ следует, что $A \cap E \neq \emptyset$. Кроме того из $(\{a\}, E) \in S$, $(\{a\}, \{b\}) \notin S$ и аксиомы (S3) вытекает, что $Y \nsubseteq E$, так как $B \in Y \setminus E$.

Пусть $(A, B) \in S$ и предположим, что $A \cap E \neq \emptyset$ и $B \nsubseteq E$. Пусть $c \in A \cap E$ и $d \in B \cap \Omega \setminus E$. Из аксиомы (S3) следует, что $(\{c\}, \{d\}) \in S$, а из аксиомы (S1) вытекает, что $(\{c\}, \{c\}) \in S$. Согласно аксиоме (S5) и $(\{a\}, E) \in S$ имеем, что $(\{a, c\}, \{c\}) \in S$, а используя аксиомы (S3) получаем, что $(\{a\}, \{c\}) \in S$. Из $(\{a\}, \{c\}) \in S$, $(\{c\}, \{d\}) \in S$ и из аксиомы (S2) следует, что $(\{a\}, \{d\}) \in S$. И наконец на основе (S4) имеем : $(\{a\} \cup \{a\}, E \cup \{d\}) = (\{a\}, E') \in S$, что является противоречием, так как $E' \supset E$. Таким образом $(A, B) \in S$, $A \cap E \neq \emptyset$ следует, что $B \subseteq E$. Тем самым утверждение (iii) и теорема 1.1. доказаны.

§ 2. Равенственное множество

Пусть дана реляция R над множеством Ω.

Определение 2.1. Определим равенственное множество ε_R реляции R следующим образом:

Пусть h, g $\in$ R, введем обозначение

$$E(h, g) = \{a\in\Omega : h(a) = g(a)\} ;$$

Тогда

$$\varepsilon_R = \{E(h, g) : h, g\in R, \quad h \neq g\} .$$

Определение 2.2. Пусть дано семейство подмножеств $\mathbf{A}$ над множеством Ω. Назовем $\mathbf{A}$ Δ-системой, если для любых множеств A, B, C, D (A, B, C, D $\subseteq \Omega$) таких что A $\neq$ B и C $\neq$ D, выполняется соотношение $A \cap B = C \cap D$.

Замечание: легко заметить, что семейство множеств $\mathbf{A}$ является Δ-системой тогда и только тогда, когда для любых A, B ($A\neq B$, A, B$\in\mathbf{A}$) $A\cap B = \cap\mathbf{A}$.

Теорема 2.1. Пусть дана реляция R над множеством Ω .

(i) если h, f, g$\in$R , то {E(h, g), E(h,f), E(g, f)} является Δ-системой

(ii) если $\varepsilon = \{E_{ij} : 1 \leq i < j \leq k\}$ такое, что для каждого i, j, ℓ $(1 \leq i < j < \ell \leq k)$ семейство $\{E_{ij}, E_{i\ell}, E_{j\ell}\}$ является Δ-системой, то существует такая реляция R для которой $\varepsilon = \varepsilon_R$.

Доказательство.

(i) Из-за симметрии достаточно показать, что $a\in E(h,g) \cap E(g,f) \to a\in E(h,f)$. Действительно,

$a \in E(h,g) \cap E(g,f) \Longleftrightarrow h(a) = g(a) = f(a) \rightarrow$

$\rightarrow h(a) = f(a) \Longleftrightarrow a \in E(h,f).$

(ii) Для докательства достаточно сконструировать по данному семейству ε такую реляцию R, что $\varepsilon = \varepsilon_R$, причем реляция R должна содержать ровно k элементов, т.е. $R = \{h_1, h_2, \dots, h_k\}$.

Конструкцию проведем индуктивно. Пусть дан произвольный элемент $h_1 \in R$, и предположим, что мы уже определили элементы $h_1, h_2, \dots, h_n$ так, что если $1 \le i < j \le n$, то $E(h_i, h_j) = E_{ij}$, где $n < k$.

Пусть

$$h_{n+1}(a) = \begin{cases} h_i(a), \text{ если существует такое } i (1 \le i \le n), \text{ что } a \in E_{i,n+1} \\ \max(h_i(b) : b \in \Omega \quad i \le 1 \le n) + 1 \text{ иначе.} \end{cases}$$

Тогда

(a) h_{n+1} является хорошо определенным элементом. Мы покажем, что если $a \in E_{i,n+1} \cap E_{j,n+1}$ $(i,j \le n)$, то $h_i(a) = h_j(a)$. Это действительно так, поскольку семейство множеств $\{E_{i,j}, E_{i,n+1}, E_{j,n+1}\}$ по предложению является Δ-системой. Таким образом

$a \in E_{i,n+1} \cap E_{j,n+1} \rightarrow a \in E_{i,j} = E(h_j, h_j)$

по индуктивному предположению.

(b) если $1 \le i \le n$ и $a \notin E_{i,n+1}$, то $h_i(a) \ne h_{n+1}(a)$.

Если

$a \notin \bigcup_{j=1}^{n} E_{j,n+1}$, то $h_i(a) \ne h_{n+1}(a)$

по определению h_{n+1} . А если существует такое j $(1 \le j \le n)$, что $a \in E_{j,n+1}$, тогда по пункту (a) $h_j(a) = h_{n+1}(a)$. Элемент $a \notin E_{ij}$, потому что $\{E_{i,j}, E_{i,n+1}, E_{j,n+1}\}$ является Δ-системой, по этому по индуктивному предположению

$h_i(a) \neq h_j(a)$, т.е. $h_i(a) \neq h_{n+1}(a)$. Из-за пунктов (a) и (b) $E(h_i, h_{n+1}) = E_{i,n+1}$ $(1 \leq i \leq n)$, то есть индукционный шаг работет.

Пусть $R = \{h_1, h_2, \ldots, h_k\}$. Тогда легко заметить, что $\varepsilon_R = \varepsilon$. тем самым теорема 2.1. доказана.

По теореме 2.1. можно естественным образом аксиоматизировать полные семейства типа **f-d-s-w** .

Пусть даны семейства $\mathbf{F}$, $\mathbf{D}$, $\mathbf{S}$ и $\mathbf{W}$ над множеством Ω.

F′ - аксиома

Семейство $\mathbf{F}$ удовлетворяет F′-аксиоме тогда и только тогда, когда существует такое k и множество $\{E_{i,j} : 1 \leq i < j \leq k, E_{i,j} \subseteq \Omega\}$, что выполняются следующие условия:

(i) если $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$, то существуют такие i,j $(1 \leq i < j \leq k)$, что $X \subseteq E_{i,j}$ и $Y \not\subseteq E_{i,j}$;

(ii) если $(A, B) \in \mathbf{F}$ и $A \subseteq E_{i,j}$, то $B \subseteq E_{i,j}$;

(iii) для любых i, j, ℓ $(1 \leq i < \ell \leq k)$ семейство $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ является Δ- системой.

D′- аксиома

Семейство $\mathbf{D}$ удовлетворяет D′-аксиоме тогда и только тогда, когда существует такое k и множество $\{E_{i,j} : 1 \leq i < j \leq k, E_{i,j} \subseteq \Omega\}$, что выполнятся следующие условия:

(i) если $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}$, то существуют такие i, j $(1 \leq i < j \leq k)$, что $X \cap E_{i,j} \neq \emptyset$ и $Y \cap E_{i,j} = \emptyset$;

(ii) Если $(A, B) \in \mathbf{D}$ и $A \cap E_{i,j} \neq \emptyset$, то $B \cap E_{i,j} \neq \emptyset$;

(iii) То же условие, что и в пункте (iii) F'-аксиомы.

S' - аксиома.

Семейство $\mathbf{S}$ удовлетворяет S'-аксиоме тогда и только тогда, когда существует такое k и множество $\{E_{i,j} : 1 \le i < j \le k\ ,\ E_{ij} \subseteq \Omega\}$, что выполняются следующие условия:

(i) Если $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{S}$, то существуют такие i, j $(1 \leqslant i < j < k)$, что $X \cap E_{i,j} \neq \emptyset$ и $Y \notin E_{i,j} = \emptyset$;

(ii) Если $(A, B) \in \mathbf{S}$ и $A \cap E_{i,j} \neq \emptyset$, то $B \subseteq E_{i,j}$;

(iii) То же условие, что и в пункте (iii) F'-аксиомы.

W' - аксиома

Семейство $\mathbf{W}$ удовлетворяет W'-аксиоме тогда и только тогда, когда существует такое k и $\{E_{i,j} : 1 \le i < j \le k\ ,\ E_{i,j} \subseteq \Omega\}$, что выполняются следующие условия:

(i) Если $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{W}$, то существуют такие i, j $(1 \le i < j \le k)$, что $X \subseteq E_{i,j}$ и $Y \cap E_{i,j} = \emptyset$;

(ii) Если $(A, B) \in \mathbf{W}$ и $A \subseteq E_{i,j}$, то $B \cap E_{i,j} \neq \emptyset$;

(iii) То же условие, что и в пункте (iii) F'-аксиомы .

Замечание. Множество $E_{i,j}$ определенное в F'-аксиоме является максимальным элементом [1] , т.е. если $(A, B) \in \mathbf{F}$ и $A \subseteq E_{i,j}$, то $B \subseteq E_{i,j}$. Аналогичным образом можно определить максимальные элементы для полных семейств типа **d-s-w** , и легко получаются результаты, аналогичные результатам Армстронга [1] .

Теорема 2.2.

(i) Пусть $\mathbf{Y} \subseteq P(\Omega) \times P(\Omega)$ и $Y \in \{F, D, S\}$. Семейство $\mathbf{Y}$ удовлетворяет Y-аксиоме тогда и только тогда, когда удовлетворяет и Y'-аксиоме.

(ii) Для любого множества Ω ($|\Omega| \geq 3$) существует семейство $\mathbf{W}$ такое, что $\mathbf{W}$ удовлетворяет W-аксиоме и не удовлетворяет W'-аксиоме.

Доказательство.

(i) Пусть $Y = F$, $\mathbf{Y} = \mathbf{F}$ и предположим, что $\mathbf{F}$ удовлетворяет F-аксиоме. Покажем, что $\mathbf{F}$ удовлетворяет и F-аксиоме. Действительно, пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}$ тогда $E(X, Y) \subseteq \Omega$ то множество, которое гарантирует F-аксиома. Далее, пусть $\{E_2, E_3, \ldots, E_k\} = \{E(X, Y) : (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{F}\}$.

Пусть $E_{1,j} = E_j$, если $1 < j \leq k$, и $E_{i,j} = E_i \cap E_j$, если $1 < i < j \leq k$. Мы утверждаем, что тогда семейство множества $\{E_{i,j} : 1 \leq i < j \leq k\}$ показывает, что семейство $\mathbf{F}$ удовлетворяет F'-аксиоме. Очевидно, что пункт (i) F'- аксиомы выполняется.

Пусть $(A, B) \in \mathbf{F}$ и предположим, что $A \subseteq E_{i,j}$ $(1 \leq i < j \leq k)$, т.е. $A \subseteq E_j$, если $i = 1$ и $A \subseteq E_i \cap E_j$, если $i > 1$. По пункту (ii) F-аксиомы и определению множества $E_2, E_3, \ldots, E_k$ $B \subseteq E_{i,j}$. Это означает, что пункт (ii) F'-аксиомы выполняется.

Пусть $1 \leq i < j < \ell \leq k$, тогда рассмотрим два случая:

(a) Пусть $i = 1$. Тогда $E_{i,j} = E_j$, $E_{i,\ell} = E_\ell$ и $E_{j,\ell} = E_j \cap E_\ell$ поэтому пересечение любых двух множеств семейства $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ равно $E_j \cap E_\ell$. Это означает, что $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ является Δ-системой.

(в) Пусть $i > 1$. Тогда $E_{i,j} = E_i \cap E_j$, $E_{i,\ell} = E_i \cap E_\ell$
$E_{j,\ell} = E_j \cap E_\ell$ и поэтому пересечение любых двух элементов множеств $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ равно $E_i \cap E_j \cap E_\ell$.
Отсюда вытекает, что семейство $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ является Δ-системой.

Если семейство $\mathbf{F}$ удовлетворяет F'-аксиоме, то очевидно, что оно удовлетворяет и F-аксиоме. Тем самым случай $Y = F$ полностью доказаны.

Пусть $Y = D$ и $\mathbf{Y} = \mathbf{D}$, и предположим, что семейство $\mathbf{D}$ удовлетворяет D-аксиоме. Покажем, что тогда $\mathbf{D}$ удовлетворяет и D'-аксиоме.

Действительно, пусть $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}$ тогда $E(X, Y) \subseteq \Omega$, то множество, которое гарантирует D-аксиома. Далее, пусть $\{E_1, E_2, \ldots, E_k\} = \{E(X, Y) : (X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathbf{D}\}$.
Если $1 \le i \le k$, то $E_{2i-1,2i} = E_i$, а если $1 \le i < j \le 2k$ и множество $E_{i,j}$ не определено, то $E_{i,j} = \emptyset$.

Мы утверждаем, что семейство множеств $\{E_{i,j} : 1 \le i < j \le 2k\}$ показывает, что семейство $\mathbf{D}$ удовлетворяет D'-аксиоме. Действительно, легко заметить, что пункт (i) D'-аксиомы выполняется. Заметим, что если множество $E_{i,j}$ не совпадает ни с одним из множеств $E(X, Y)$, то $E_{i,j} = \emptyset$ и поэтому $((A, B) \in \mathbf{D} \,\&\, A \cap E_{i,j} \ne \emptyset \to B \cap E_{i,j} \ne \emptyset)$. Отсюда вытекает, что пункт (ii) D'-аксиомы также выполняется. А множество семейств $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ $(1 \le i < j < \ell \le k)$ является Δ-системой, потому что по крайней мере два из этих элементов пустые.

Легко заметить, что если семейство $\mathbf{D}$ удовлетворяет D'-аксиоме, то оно удовлетворяет и D-аксиоме.

В случае $Y = S$ доказательство проводится аналогично случаю $Y = F$.

(ii) Для простоты будем считать, что $\Omega = \{a, b, c\}$. (В общем случае считается, что $a, b \in \Omega$ и элемент $\{c\}$ надо заменить всеми элементами $\Omega \setminus \{a, b\}$.)
Пусть $W = \{(A, B) : A \subseteq \{a\} \to a \in B$ и $A \subseteq \{b\} \to b \in B\}$. Заметим, что тогда семейство W удовлетворяет D-аксиоме. Действительно, если $(X, Y) \in P(\Omega) \times P(\Omega) \setminus W$, то либо $X \subseteq \{a\}$ и $a \notin Y$, либо $X \subseteq \{b\}$ и $b \notin Y$. В первом случае множество $E = \{a\}$, а во втором случае множество $E = \{b\}$ показывает, что W-аксиоме выполняется.

Мы утверждаем, что для только что определенного семейства W W'-аксиома не выполняется. Доказательство проведем от противного. Предположим, что семейство множеств $\varepsilon = \{E_{i,j} : 1 \le i < j \le k\}$ удовлетворяет W'-аксиоме для семейства W. Тогда

(1) $\{a\} \in \varepsilon$ и $\{b\} \in \varepsilon$,

так как $(\{a\}, \Omega \setminus \{a\}) \notin W$ и $(\{b\}, \Omega \setminus \{b\}) \notin W$;

(2) $\emptyset \notin \varepsilon$ и $\{c\} \notin \varepsilon$,

так как $(\emptyset, \Omega) \in W$ и $(\{c\}, \Omega \setminus \{c\}) \in W$.

Рассмотрим два случая:

(a) Пусть $\{a\} = E_{i,j}$ и $\{b\} = E_{i,j}$. Тогда, поскольку $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ является Δ-системой, то либо $E_{j,\ell} = \emptyset$, либо $E_{j,\ell} = \{c\}$, а это противоречит тому, что $\{c\}, \emptyset \notin \varepsilon$ по (2).

(в) Пусть $\{a\} = E_{i,j}$, $\{b\} = E_{\ell,m}$ и $|\{i,j,\ell,m\}| = 4$.

В дальнейшем можно предположить, что $(i,j) = (1,2)$ и $(\ell,m) = (3,4)$, потому что мы изучаем случай, когда

$$\varepsilon = \{E_{i,j}, E_{i,\ell}, E_{i,m}, E_{j,\ell}, E_{j,m}, E_{\ell,m}\}.$$

/Действительно, семейство ε может содержать и другие элементы тоже./

Исследуем множество $E_{1,3}$.

Если $E_{1,3} = \{a\}$ или $E_{1,3} = \{b\}$, то мы приходим к случаю (а). Разумеется, $E_{1,3} \neq \{c\}$ и $E_{1,3} \neq \emptyset$ из-за (2). Если $E_{1,3} = \{b, c\}$, то $E_{1,2} \cap E_{1,3} = \emptyset$ и, поскольку $\{E_{1,2}, E_{1,3}, E_{2,3}\}$ является Δ-системой, то $E_{2,3} = \emptyset$, что противоречит (2). Следовательно, $a \in E_{1,3}$. Значит $a \in E_{1,3} \cap E_{1,2}$, а поскольку $\{E_{1,2}, E_{1,3}, E_{2,3}\}$ - Δ-система, то $a \in E_{2,3}$.

С другой стороны $a \notin E_{3,4}$ и значит $a \notin E_{3,4} \cap E_{2,3}$. Поскольку $\{E_{2,3}, E_{3,4}, E_{2,4}\}$ является Δ-системой и $a \in E_{2,3}$, то $a \notin E_{2,4}$. Отсюда следует, что $E_{2,4} \subseteq \{b,c\}$. В то же время $E_{2,4} \neq \{b\}$, так как это приводит к случаю (а). Кроме того, $E_{2,4} \neq \emptyset$ и $E_{2,4} \neq \{c\}$ из-за (2). Следовательно, $E_{2,4} = \{b,c\}$. Отсюда получаем, что $b \in E_{2,4} \cap E_{3,4}$ и значит $b \in E_{2,3}$. Таким образом $\{a,b\} \subseteq E_{2,3}$. Поскольку $b \notin E_{1,2}$, то отсюда следует, что $b \notin E_{1,3}$. Сравнивая этот результат с полученными выше ограничениями, получаем, что $E_{1,3} = \{a,c\}$. Но тогда $E_{1,3} \cap E_{3,4} = \emptyset$ и $E_{1,3} \cup E_{3,4} = \Omega$, то есть $E_{1,4} = \emptyset$, что удовлетворяет утверждению (2).

Замечание: Теорема 2.2. указывает на существующую разницу между слабыми и всеми другими зависимостями. Дело в том, что если первое множество некоторой зависимости пустое, то это представляет проблему только для слабых зависимостей.

Теорема 2.3. Пусть дано семейство $\mathbf{Y} \subseteq P(\Omega) \times P(\Omega)$ и $\mathbf{Y} \in \{F, D, S, W\}$. Если семейство $\mathbf{Y}$ удовлетворяет Y'-аксиоме, то существует реляция R над множеством Ω такая, что $\mathbf{Y} = \mathbf{Y}_R$. Наоборот: если дана реляция R над множеством Ω, то семейство $\mathbf{Y}_R$ удовлетворяет Y'-аксиоме; где $Y' \in \{F', D', S', W'\}$.

Доказательство. Покажем, что семейство $\mathbf{Y}$ удовлетворяет F'-аксиоме, а это проверяется на множестве $\varepsilon = \{E_{i,j} : 1 \le i < j \le k\}$. Тогда из-за пункта (iii) Y'-аксиомы и из-за пункта (ii) теоремы 2.1. существует такая реляция R над Ω, что $\mathbf{Y} = \mathbf{Y}_R$.

Естественно, что из-за пунктов (i) и (ii) Y'-аксиомы $\mathbf{Y} = \mathbf{Y}_\mathbf{R}$.

Наоборот, если дана реляция R над множеством Ω , то семейство множеств $\varepsilon_R = \{E_{i,j} : 1 \le i < j \le k\}$ указывает на то, что семейство $\mathbf{Y}_R$ действительно удовлетворяет Y'- аксиоме, где $R = \{h_1, h_2, \dots, h_k\}$ и $E_{i,j} = E(h_i, h_j)$. Тем самым теорема доказана.

ЛИТЕРАТУРА

[1] Armstrong, W.W., Dependency structures of data base relationships, Information Processing 74, North-Holland Publ.Co.,(1974) 580-583.

[2] Armstrong, W.W., On the generation of dependency structures of relational data bases, Publication #272, Université de Montréal.(1977).

[3] Beeri, C. - Fagin, R. - Howard, J.H., A complete axiomatization for functional and multivalued dependencies in database relations, Proc. ACM SIGMOD Int.Conf. on Management of Data, Toronto (1977) 47-61.

[4] Békéssy, A. - Demetrovics, J., Contribution to the theory of data base relations, Discrete Math. 27(1979) 1-10.

[5] Békéssy, A. - Demetrovics, J. - Hannák, L. - Frankl, P. - Katona, G., On the number of maximal dependencies in a data base relation of fixed order, Discrete Math. 29 (1980).

[6] Codd, E.F., A relational model of data for large shared data banks, Comm. ACM. 13(1970) 377-387.

[7] Codd, E.F., Further normalization of the data base relational model, Courant Computer Science Symposia 6 Data Base System, Prentice Hall, Englewood Cliffs, N.J. (1971) 33-64.

[8] Czédli, G., Függőségek relációs adatbázis modellben, Alk. Mat. Lapok (1980)

[9] Demetrovics, J., Candidate keys and antichains, SIAM J. on Algebraic and Discrete Methods 1 (1980) 1,92.

[10] Demetrovics, J., On the equivalence of candidate keys with Sperner systems, Acta Cybernetica 4 (1979) 247-252

[11] Fagin, R., Multivalued dependencies and a new normal form for relational databases, ACM Trans. on Database System 2:3 (Sept. 1977), 262-278.

[12] Rissanen, J., Theory of relations for databases - a tutorial survey, Proc. 7th Symp. on Foundations of Computer Science, Poland 1978, Lecture Notes in Computer Science 64, Springer Verlag, pp. 537-551.

[13] Sagiv, Y., An algorithm for inferring multivalued dependencies that works also for a subset of propositional logic, Department of Computer Science, University of Illinois at Urban-Champaign, 1979.

ÖSSZEFOGLALÁS

A relációs adatmodell funkcionális függőségeinek általánositása

Demetrovics J. - Gyepesi Gy.

Ebben a cikkben relációs adatmodellek funkcionális függőségeivel és ennek lehetséges általánositásával; a duális, erős és gyenge függőségekkel foglalkozunk. Axiomatizáljuk ezen függésfajták teljes családjait /a funkcionális függőségek teljes családjaira [1]-ben adtak először axiómarendszert/.
Axiómáink mátrixok egyenlőséghalmazának egy kombinatórikus jellemzésén alapulnak. Rámutatunk egy a gyenge függés és a többi között fennálló, fontos különbségre.

A generalisation of the functional dependencies in the relational data base

Demetrovics, J. - Gy. Gyepesi,

In this paper we deal with the functional dependencies of relational date models and their three generalizations: the dual, strong and weak dependencies. We axiomatize the full families of these dependencies of any kind /full families of functional dependencies were firstly axiomatized in [1] and that of dual and strong dependencies in [8] as well; the axiomatization of full families of weak dependencies is a new result/.

Our axioms are based on a combinatorial characterization of equality-sets of matrices. We prove an important difference between the weak dependencies and the rest.

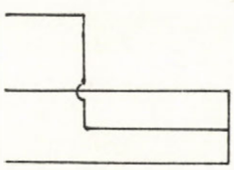
MTA Számitástechnikai és Automatizálási Kutató Intézete, Közlemények 24/1980.

Result of Cluster Analysis: Dendrogram
/An Algorithm for Displaying the Dendrogram/

Ferenc Juhász - Károly Tóth
Budapest

Many of the methods of cluster analysis consists in hierarchical procedures. The results of these procedures are contained most concisely in the so-called dendrogram.

The dendrogram is a binary tree the branches of which correspond to the mergers /separations/ of clusters. The dendrogram of several methods /Centroid, Median, Eigen.../ may have reversal i.e. it may contain the following isomorphic tree:



An important auxliary tool of the high-level software is the plotter. Displaying the dendrogram by means of the plotter is justified by two points of view:

(1) the dendrogram obtained can be published without any alteration,

(2) the reversal phenomenon is easy to handle.

It is difficult to get literary data on programs used for displaying dendrograms on a plotter [1, p.328], therefore it appeared desirable to develop the subroutines described below.

In order that the results obtained by different methods may be compared, the suitable version must be selected from the possible 2^{N-1} isomorphic variants with the use of some uniform principle. It is practical to choose such a relatively natural permutation of the elements that the dendrogram plotted on its order does not contain any intersection if not necessary.

The question arises what is the minimum information serving for describing the dendrogram.

(1) The hierarchical procedure generally constitues a series $(I(K), J(K))$, $K=1,\dots,N-1$ which expresses that in step K the clusters I(K) and J(K) merge. /The algorithm identifies each cluster with the element of the lowest order occurring in it./

(2) It is natural that the clusters contain the elements ordered according to their serial number.

The above informations determine the permutation of the dendrogram elements univocally. This permutation is produced by the subroutine PERMUT. The dendrogram is displayed by the subroutine DENDPLOT called after this.

The subroutines pertaining to the concrete machine realization used in the program, i.e. the subroutines PLOTOPLEN, PLOTOLOS, MOVE, DRAW and PLOTVAL have been written for the computer CDC 3300 and they control the plotter CALCOMP 363. The first two serve for opening and closing the plotterfile, the other two for moving the pen, while the last one is a subroutine for writing characters.

Figures 1 and 2 show dendrograms concerning to the problem published in paper [2] /based on Bernstein's method/.

SUBROUTINE PERMUT (I,J,N,NL,IPERM,IL,JL,NEXT)

Formal parameters

I	integer array	(NL)	input:	in step K clusters I(K)
J	integer array	(NL)	input:	and J(K) merge
N	integer		input:	number of points
NL=N-1			input	
IPERM	integer array	(N)	output:	permutation of the serial numbers of the elements
IL	integer array	(NL)	workspace	
JL	integer array	(NL)	workspace	
NEXT	integer array	(N)	workspace	

SUBROUTINE DENDPLOT (C,H,I,J,IPERM,N,NL,S,PX,PY)

Formal parameters

C	real		input:	horizontal dimension of the dendrogram (in 0.01 inches)
H	real		input:	height of the characters (in 0.01 inches)
I	integer array	(NL)	input:	as above
J	integer array	(NL)	input:	as above
IPERM	integer array	(N)	input:	as above
N	integer		input:	as above
NL=N-1			input	
S	real array	(NL)	input:	levels belonging to the mergers
PX	real array	(N)	workspace	
PY	real array	(N)	workspace	

External subroutines

PLOTOPEN (IDSI,IBLK)

IDSI	integer	input: identification of the plotterfile
IBLK	integer	input: maximum number of blocks

PLOTCLOS

MOVE (X,Y)

X	real	input: drawing pen in lifted position
Y	real	input: is moving into point (X,Y)

DRAW (X,Y)

X	real	input: pen is drawing a line from the
Y	real	input: actual point into point (X,Y)

PLOTVAL (IFORMAT,IVALUE,NCHAR,HEIGHT,ANGLE)

IFORMAT	integer array	input: format required for drawing the value of IVALUE
IVALUE	integer	input: value to be drawn
NCHAR	integer	input: number of characters to be drawn
HEIGHT	real	input: height of numbers (in 0.01 inches)
ANGLE	real	input: rotation angle of characters (in radians)

```
      SUBROUTINE PERMUT(I,J,N,NL,IPERM,IL,JL,NEXT)
C
C        CONSTRUCT THE PERMUTATION OF THE NODES
C
      DIMENSION I(NL),J(NL),IPERM(N)
      DIMENSION IL(NL),JL(NL),NEXT(N)
C
C        INITIALIZE IL, JL, NEXT
C
      DO 1 K=1,N
      NEXT(K)=0
    1 CONTINUE
      DO 2 K=1,NL
      IK=I(K)
      JK=J(K)
      IL(K)=NEXT(IK)
      JL(K)=NEXT(JK)
      NEXT(IK)=K
      NEXT(JK)=K
    2 CONTINUE
      DO 3 K=1,NL
      NEXT(K)=N
    3 CONTINUE
      DO 4 K=1,NL
      LK=IL(K)
      IF(LK.NE.0) NEXT(LK)=K
      LK=JL(K)
      IF(LK.NE.0) NEXT(LK)=K
    4 CONTINUE
C
C        GO OVER THE TREE
C
      DO 5 IK=1,NL
      IF(I(IK).NE.1) GO TO 5
      K=IK
      GO TO 6
    5 CONTINUE
    6 IST=0
   11 IF(IL(K).NE.0) GO TO 12
      IST=IST+1
      IPERM(IST)=I(K)
      IL(K)=-1
   12 IF(IL(K).EQ.-1) GO TO 21
      KN=IL(K)
      IL(K)=-1
      K=KN
      GO TO 11
   21 IF(JL(K).NE.0) GO TO 22
      IST=IST+1
      IPERM(IST)=J(K)
      JL(K)=-1
   22 IF(JL(K).EQ.-1) GO TO 31
      KN=JL(K)
      JL(K)=-1
      K=KN
      GO TO 11
   31 K=NEXT(K)
      IF(K.EQ.N) GO TO 41
      GO TO 11
   41 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DENDPLOT(C,H,I,J,IPERM,N,NL,S,PX,PY)
C
C        KNOWING THE PERMUTATION OF THE NODES DISPLAY THE DENDROGRAM
C
      DIMENSION I(NL),J(NL),IPERM(N),S(NL)
      DIMENSION PX(N),PY(N)
      DIMENSION IFORMAT(1)
C
C        INITIALIZE THE PARAMETERS
C
      IFORMAT(1)=4H(I3)
      OX=90.
      OY=90.
      H67=H*6./7.
      H97=H*9./7.
      PI=3.1415926
      PI2=PI/2.
      SMAX=0.
      DO 4 K=1,NL
      IF(S(K).GT.SMAX) SMAX=S(K)
    4 CONTINUE
      CALL PLOTOPEN(11,500)
C
C        DISPLAY THE CHARACTERS
C
      X=OX
      Y=OY-3.*H67
      DO 1 K=1,N
      X=X+H97
      CALL MOVE(X,Y)
      L=IPERM(K)
      CALL PLOTVAL(IFORMAT,L,3,H,PI2)
    1 CONTINUE
C
C        DRAW THE TREE
C
      DO 2 K=1,N
      L=IPERM(K)
      PX(L)=OX+H67+H97*(K-1)
      PY(L)=OY
    2 CONTINUE
      DO 3 K=1,NL
      IK=I(K)
      JK=J(K)
      BX=PX(IK)
      BY=PY(IK)
      SY=OY+C*S(K)/SMAX
      EX=PX(JK)
      EY=PY(JK)
      CALL MOVE(BX,BY)
      CALL DRAW(BX,SY)
      CALL DRAW(EX,SY)
      CALL DRAW(EX,EY)
      PX(IK)=(BX+EX)/2.
      PY(IK)=SY
    3 CONTINUE
      CALL PLOTCLOS
      RETURN
      END
```

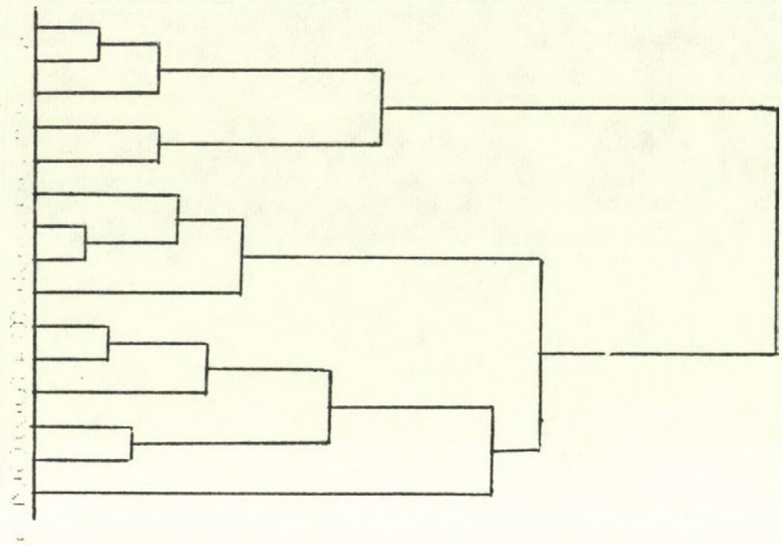


Figure 1 Furthest Neighbour

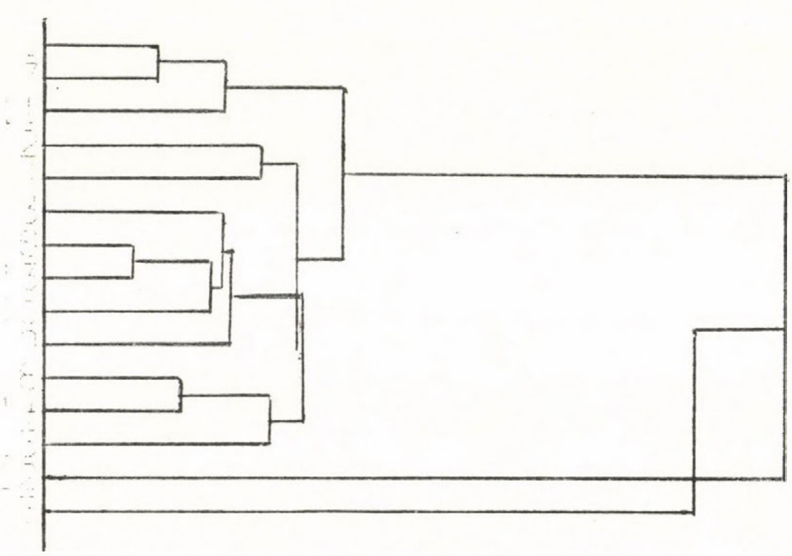


Figure 2 Median

REFERENCES

[1] Anderberg,M.R. /1973/: Cluster Analysis for Applications. Academic Press, New York - San Francisco - London.

[2] Juhász,F. /1978/79/: The Analysis of the Distribution of Blood Groups by Means of Clustering Methods. Mitt.Arch.Inst. 8/9.

ÖSSZEFOGLALÁS

Juhász F. - Tóth K.

A dolgozat algoritmust közöl dendogram plotteren való megjelenitésére.

Результат кластерного анализа: дендорограмма

Юхас Ф. - Тот К.

В работе дается алгоритм для изображения дендрограммы на плоттере.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 24/1980.

PARABOLA ÉS ELLIPSZIS KÖZELÍTÉSE EGYENESSZAKASZOKKAL ÉS KÖRÍVEKKEL

Szepesvári István

A numerikus szerszámgépek automatikus irányitásánál fellép különböző görbéknek egyenesszakaszokkal, illetve körivekkel való közelitésének problémája. Ebben a dolgozatban parabola ($y=ax^2$ alaku) és ellipszis ($(\frac{x}{a})^2 + (\frac{y}{b})^2 = 1$ alaku) közelitésével foglalkozunk. FORTRAN nyelven készültek különböző fajta közelitésekre programok (szubrutinok). A programok eredményül a közelitő alakzat (pl. törött vonal) bizonyos pontjait adják ki, és természetesen azt, hogy két szomszédos pont között egyenesszakaszt vagy körivet kell-e huzni (és a köriv középpontját is megkapjuk). Megfelelő transzformációval általános egyenletü parabola és ellipszis is approximálható. Igy az eredmények speciális alkatrészprogramozó nyelvet használó programokban alkalmazhatók.

Kezdve egyszerü közelitésektől egyre bonyolultabb szubrutinokkal egyre kevesebb szakasszal (ivvel) valósitunk meg ugyanolyan pontosságu közelitéseket.

1. PARABOLA KÖZELITÉSE EGYENESSZAKASZOKKAL

SUBROUTINE PAR1SYM (X1,X2,YUE,YUU,A,EPS,XV,IE,IU,M4,IER)

Az $y = ax^2$ parabolát (illetve annak egy P_1(X1,Y1), P_2(X2,Y2) zárt ivét) közeliti minimális számu egyenesszakasszal oly módon, hogy az egymáshoz kapcsolódó szakaszok az y-tengelyre nézve tükrösen helyezkednek el, és az $y = ax^2 + \varepsilon$ és $y = ax^2 - \varepsilon$ parabolák által meghatározott sávból nem lépnek ki, és a $(0, -\varepsilon)$ pontban töréspont van. (1. ábra)

Paraméterek:

X1 : az iv kezdőpontjának abszcisszája (X1 > X2 legyen mindig)

X2 : az iv végpontjának abszcisszája

YUE: a közelitő alakzat kezdőpontjának ordinátája

YUU: a közelitő alakzat végpontjának ordinátája

A : az $y = ax^2$ parabola esetén $A = a$

EPS: hibahatár

XV : tömb, amely legalább az X2-től a -X2 abszcisszájáig tartalmazza a parabolát közelitő alakzat töréspontjainak abszcisszáját és ordinátáját, felváltva az 1 és M4 közötti indexü rekeszekben ($M4 \leq 400$)

IE

IU az eredmény kiiratásához szükségesek (később következik)

M4

IER: hibaparaméter (0 esetén a futás helyes)

X1, X2, A, EPS bemenő, a többi kimenő paraméter.

A szubrutin hivása:

CALL PAR1SYM(X1,X2,YUE,YUU,A,EPS,XV,IE,IU,M4,IER)

Példa az eredmények kiiratására:

```
PRINT 17,X1,YUE
IUU=IU-2
IF(IE.EQ.IUU) GO TO19
DO 18 IJ=IE,IU,2
XKO=XV(IJ)
```

```
YKO=XV(IJ+1)
18 PRINT 17, XKO,YKO
19 PRINT 17,X2,YUU
17 FORMAT (1H,2E15.7)
```

A szubrutin használja a PAEGER1, PAEGME1 szubrutinokat.

Példa: X1 = 7.5
X2 =-3.9
A = 0.3
EPS = 0,05

Eredményként 11 egyenesszakasszal közelit.

Y1-et (ill. Y2-öt) a töröttvonal és az x=X1 (ill. x=X2 egyenes metszéspontjaként kapjuk.)

SUBROUTINE PAR1(X1,X2,A,EPS,XY,L,IER)

A PAR1SYM-től abban különbözik, hogy általában nem az y-tengelyre szimmetrikus módon közelit. (P_1 és P_2 most mindenképp az $y = ax^2 - \varepsilon$ parabolán van).

Az eredményt az XY tömbben kapjuk, 1-től M4-ig terjedő indexü rekeszekben, hasonlóan, mint a PAR1SYM-ben.

Rövidebb parabolaivek esetén gazdaságosabban használható - helyigény és gépidő tekintetében - mint a PAR1SYM. Egyébként a paraméterek értelmezése ugyanaz mint a PAR1SYM-nél.

Felhasználja a PAEGME1, PAEGER1 szubrutinokat.

Próba: X1 = 7.5
X2 = -3.9
A = 0.3
EPS = 0.05

Eredményként 10 egyenesszakasszal közelit.

Legyen $P_1(x_1,y_1)$ az $y = ax^2 - \varepsilon$ parabola (p_1) egyik futó pontja. Vegyünk a p_1 P_1-beli normálisának az $y = ax^2 + \varepsilon$ parabolával (p_2) való (P_1-hez közelebb eső) metszéspontját, P_2-öt. A P_1P_2 szakasz a p_1 és p_2 parabola P_1-beli merőleges távolsága, (2. ábra). Ismeretes, hogy a P_1 P_2 távolság Y_1 növekedésével monoton csökken. Célunk, hogy a közelitő töröttvonal az $y = ax^2$ parabolától merőlegesen mért $\pm\varepsilon$ távolságra levő sávban legyen. Alkalmazhatnánk tehát a PAR1-beli egyszerü módszert. Ehhez a módszerhez képest azonban általában csökkentjük a közelitő szakaszok számát a következő módon. (2. ábra)

(A PARMESYM-ben és PARME-bem ezt használjuk). A P_1-ből P_2-höz huzott érintő (e) P_1-et a P_1'-ben metszi. A P_1'-beli normális metszéspontja p-vel ($y = ax^2$) a p_1 és p_2 közötti sávban legyen U. A p-től U-ban merőlegesen mért $\pm\varepsilon$ távolságra legyen a $p_1'(y = ax^2 - \varepsilon')$ és a $p_2'(y = ax^2 + \varepsilon')$ parabola ($\varepsilon' > \varepsilon > 0$). Most a P_1'-ből a p_2'-hoz nuzott erintő (e) a p_1'-öt a P_1''-ben metszi. A P_1'-ben alkalmazott eljárást P_1'' ben ismételjük, stb. Nyilvánvaló, hogy igy gazdaságosabb módszerhez jutunk.

SUBROUTINE PARMESYM (X1,X2,YUE,YUU,A,EPS,XV,IE,IU,M4,IER)

Az PAR1SYM-től abban különbözik, hogy figyelembe veszi az $y = ax^2$ parabolára merőlegesen mért $\pm\varepsilon$ távolságot, tehát az ottaninál szélesebb sávban, általában kevesebb elemmel közelit a módszer.

Felhasználja a PAEGME1, PAEGER1 szubrutinokat.

Paraméterek: mint PAR1SYM-nél.

Példa: X1 = 7.5

X2 = -3.9

A = 0.3

EPS = 0.05

Eredményként 8 egyenesszakasszal közelit.

SUBROUTINE PARME (X1,X2,A,EPS,XY,L,IER)

A PARMESYM-től abban különbözik, hogy általában nem az y-tengelyre szimmetrikusan közelit.

Az eredményt az XY tömbben kapjuk, az 1-től M4-ig terjedő indexü rekeszekben.

Paraméterek: mint PAR1SYM-nél.

Felhasználja a PAEGME1, PAEGER1 szubrutinokat.

```
Példa:    X1 =  7.5
          X2 = -3.9
          A  =  0.3
         EPS =  0.05
```

Eredményként 8 egyenesszakasszal közelit.

2. PARABOLA KÖZELITÉSE EGYENESSZAKASZOKKAL ÉS KÖRIVEKKEL

SUBROUTINE PEKER (XV,XUR,M4,IER,IU,A)

Az egyenesszakaszokból álló közelitő törött vonalat (t) kikerekiti - két szomszédos egyenesszakaszt érintő köriv segitségével - a töréspontok közelében. Az, hogy a köriv hogyan illeszkedik a szakaszokhoz a 3. ábrából és a hozzá füzött magyarázatból kitünik. Igy a kapott közelitő görbeiv minden belső pontjában differenciálható lesz. Feltesszük, hogy egyik szakasz sem tul rövid, és a hibahatár is elegendően kicsi. Ez a bemenő paraméterek megfelelő megválasztásával elérhető.

Paraméterek:

XV: tömb, amely a t töröttvönal az $y = ax^2$ parabolát érinti egymásután következő pontjainak koordinátáit (abszcissza és ordinata felváltva) tartalmazza

XUR: tömb, az eredményt tartalmazza a következő elrendezésben:

XUR(1), XUR(2): (az első) egyenesszakasz (e_1) kezdőpontja

XUR(3), XUR(4): (az első) egyenesszakasz (e_1) végpontja, ami egyuttal a hozzá érintőlegesen csatlakozó k_1 köriv kezdőpontja is

XUR(5), XUR(6): k_1 középpontjának koordinátái

XUR(7), XUR(8): k_1 végpontja, ami egyuttal a hozzá csatlakozó egyenesszakasz kezdőpontja. A továbbiakban hasonló az elrendezés mint XUR(1)-től számitva, stb.

M4: az XUR tömbnek az az indexe, amelyben a közelitő görbe végpontjának ordinátája található. (M4 $\leq$ 400)

IU: bemenő paraméter; IU=M4 PARISYM és PARMESYM után; IU=M4-2 PAR1 és PARME után használva, ahol M4 értéke az egyes szubrutinokban kapott M4.

A szubrutin azonnal alkalmazható a PAR1 és PARME szubrutinok után. A PAR1SYM és PARMESYM után is alkalmazható, ha a kapott töröttvonal koordinátáit (X1,YUE,XY(3),...,X2,YUU) megfelelően helyezzük el az XV tömbben.

A szubrutin használja a SZOGFEL, PAEGME2, EGEGME és EGEGM2 szubrutinokat.

Példa: X1 = 7.5
X2 = -3.9
A = 0.3
EPS = 0.05

Eredményként 11 egyenesszakasszal és 10 körivvel közelit a PAR1SYM után használva;

Eredményként 8 egyenesszakasszal és 7 körivvel közelit a PARMESYM után használva;

Eredményként 10 egyenesszakasszal és 9 körivvel közelit a PAR1 után használva;

Eredményként 8 egyenesszakasszal és 7 körivvel közelit a PARME után használva.

3. PARABOLA KÖZELITÉSE KÖRIVEKKEL

Kézenfekvő, hogy az 1.-ban kapott egyenesszakaszokat körivekkel helyettesitsük. (4. ábra, amelyből a módszer lényege leolvasható.) Gazdaságosabbá tehetjük azonban a körivekkel való közelitést a következő módon (5. ábra). Tegyük fel, hogy a P-ből kiinduló közelitő körivet akarjuk megszerkeszteni ugy, hogy az a p_o-tól merőlegesen $\pm\varepsilon$ távolságban levő sávban legyen. P és az $y = ax^2$ parabola (p_o) merőleges távolsága $\varepsilon = \overline{UP}$, a P-n átmenő parabola (p_1) $y = ax^2 - \varepsilon_1$. Vegyünk egy $y = ax^2+\varepsilon_2$ (p_2) parabolát ($\varepsilon_2 > \varepsilon_1$). Huzzunk P-ből p_2-höz érintőt (e), ez metssze p_1-et P_1-ben. Az O középpontu kört a 4. ábra szerint megszerkesztjük, ennek PP_1 ive. A PP_1 iv a p_1 és p_2 között levő sávnál keskenyebb sávba is nyilván belefér. E célból p_2-öt p_o-hoz közelitjük a következő módon. Egy olyan p_2' parabolát szerkesztünk, amely a PMP_1 törött vonalat ($\overline{PM}$ és $\overline{MP}_1$ szakaszból áll) érinti. A p_2' merőleges távolsága az U ponttól legyen ε'. Ezt az utóbbi, p_2'-öt meghatározó eljárást (p_2 szerepét p_2' tölti be) - a kétoldali közelitést alkalmazva - többször ismételve ε' értékét be tudjuk ugy állitani, hogy $|\varepsilon'|\leq\varepsilon_p$ legyen, ahol ε_p a hibahatár.

SUBROUTINE PAR2 (X1,X2,A,EPS,XYP,L,IER)

Az $y = ax^2$ parabolát (p), illetve annak egy ivét közeliti körivekkel oly módon, hogy az egymáshoz kapcsolodó ivek a p-től merőlegesen mért $\pm\varepsilon$ távolságon belül maradnak. Az eredményt az XYP tömbben kapjuk a következő elrendezésben:

XYP(1), XYP(2): az első köriv kezdőpontjának x és y koordinátája

- -

XYP(3), XYP(4): az első köriv végpontja, ami a második köriv kezdőpontja

XYP(5), XYP(8): a második köriv végpontja

XYP(9), XYP(10): a második köriv körének középpontja

XYP(11), XYP(12): a harmadik köriv végpontja, stb.

...

XYP(L)-ben található az utolsó köriv köre középpontjának ordinátája.

Ha X1 = O (egyébként X1 > X2 mindig), majd a kapott közelités X koordinátáit ellenkező előjellel is figyelembe vesszük, akkor y tengelyre szimmetrikus közelitést kapunk.
Ha valahonnan kezdve a körivek sugara tulságosan nagy, akkor a PARME illetve PEKER szubrutinokat alkalmazhatjuk ettől a ponttól kezdve.
A szubrutin felhasználja a PONTOS, PARR, ez utóbbi pedig a PAEGME1, PAEGME2, PAEGER1, KOEGME1 szubrutinokat.

Példa: X1 = 7.5
X2 = -3.9
A = 0.3
EPS = 0.05

Eredményként 6 körivvel közelit.

SUBROUTINE PKKER (XYP,L,XYV,IER,Z)

A körivekből álló közelitő töröttvonalat (t) kikerekiti - két szomszédos körivet megfelelően érintő köriv segitségével - a töréspontok közelében, (6. ábra) a 2. ponthoz hasonlóan. Igy a kapott közelitő görbeiv minden belső pontjában differenciálható lesz. Feltesszük, hogy egyik köriv sem tul rövid, és a hibahatár is elegendően kicsi. Ez a bemenő paraméterek megfelelő választásával elérhető.

Paraméterek:

XYP : tömb, amely a t töröttvonal adatait tartalmazza (mint PAR2-ben az XYP tömb)

XYV : tömb 1-től LK-ig (ahol LK=2*L+2), amely az eredményt tartalmazza az ábra szerinti elrendezésben. Az XYV (LK) rekesz tartalmazza az utolsó köriv köre középpontjának ordinátáját.

Z : az $y = ax^2$ parabola esetén Z=a

A szubrutin a PAR2 szubrutin után alkalmazható.
Felhasználja a SZOGFEL, PAEGME2, KOEGME1 szubrutinokat.

Példa: X1 = 7.5
X2 = -3.9
A = 0.3
EPS = 0.05

Eredményként 11 körivvel közelit.
Láthatjuk, hogy az elkészült szubrutinok közül a legutóbbi adja a legkevesebb körivvel való közelitését ugyanazon parabolaivnek adott megkivánt pontosság esetén, és a közelitő görbeiv minden pontjában differenciálható.

A közelités pontosságát fokozhatjuk, ha a parabola csucsához közel körrel közelitünk a 7. ábrán bemutatott módon, ezután pedig az eddig ismertetett módszereket alkalmazzuk. Ugyanis ha a kör középpontja az y tengelyen van ($y = ax^2 \pm \varepsilon$ alaku parabolák esetén) akkor a kör és a 2 parabola érintési pontjainak meghatározására 2-odfokura redukálható egyenletet kapunk.

4. ELLIPSZIS KÖZELITÉSE EGYENESSZAKASZOKKAL

Ellipszis közelitésére a parabolára bemutatott eljárásokat - kisebb változtatásokkal - mind alkalmazhatjuk.

SUBROUTINE ELI1 (X1,Y1,X2,Y2,A,B,EPS,XUR,M4,IER,I,JEL,XV)

Az $(\frac{x}{a})^2 + (\frac{y}{b})^2 = 1$ ellipszist (illetve annak egy P_1(X1,Y1), P2(X2,Y2) zárt ivét) közeliti minimális számu egyenesszakaszszal oly módon, hogy az egymáshoz kapcsolódó egyenesszakaszok az y-tengelyre nézve tükrösen helyezkednek el, és azok az $(\frac{x}{a})^2 + (\frac{y}{b})^2 = (1 + \frac{\varepsilon}{b})^2$ és az $(\frac{x}{a})^2 + (\frac{y}{a})^2 = (1 - \frac{\varepsilon}{b})^2$ ellipszisek által meghatározott sávból (S) nem lépnek ki. (Ha a közelitő töröttvonal (t) az ellipszis (0,-b) pontjánál is közelit, akkor a (0,-b-ε) pontban van töréspont; és ha a (-a, 0) illetve (a, 0) pontjánál is közelit, akkor az $x = a + \varepsilon$ illetve $x = a - \varepsilon$ egyenesek egy szakasza része t-nek).

Paraméterek:

X1, Y1 az iv kezdőpontjának (P_1) koordinátái ($|X1|<a-\varepsilon$)

X2, Y2 az iv végpontjának (P_2) koordinátái ($|X2|<a-\varepsilon$).

Fontos, hogy a P_1-től P_2 felé tartó irány az óramutató járásával ellenkező legyen.

A, B	:	az ellipszis tengelyei
EPS	:	hibahatár
XUR	:	tömb, amelyben az eredményt kapjuk, az 1-től I+1-ig terjedő indexü rekeszekben
M4	:	a teljes ellipszist közelitő pontok kétszeres száma
IER	:	hibaparaméter, 0 esetén a futás helyes
I	:	XUR-nél volt
JEL	:	1 ha a közelitendő iv tartalmazza a (0, -b) pontot, 0 egyébként

XV : tömb, amely az 1-től M4-ig terjedő indexü rekeszekben a teljes ellipszist közelitő töröttvonal töréspontjainak koordinátáit tartalmazza.
Ezt a tömböt - ellenőrzésképpen - ajánlatos kiiratni, mert igy hiba esetén a bemenő X1, Y1; X2, Y2 értékek ujbóli megadása nem okoz gondot.
Legyen az XV tömb által megadott töröttvonal két egymásutáni pontja (XS1, YS1) és (XS2, YS2). Az (X1,Y1)-et (illetve (X2, Y2)-öt) ugy kell megadni, hogy X1 (ill. X2) az XS1 és XS2; Y1(ill. Y2) az YS1 és XS2 közé essen.

Az X1, Y1, X2, Y2, A, B, EPS, JEL bemenő paraméterek.

A szubrutin használja az ELEGER1, ELEGME1 szubrutinokat.

Példa: X1 = -1., Y1 = -29.8
X2 = 1. Y2 = +29,8
JEL = O; A = 20., B = 30., EPS = 0.03, ekkor 50 db szakasz közelit.

5. A FELHASZNÁLT SZUBRUTINOK RÖVID ISMERTETÉSE

SUBROUTINE PAEGER1 (X1,Y1,B,C,R,IER)
A P(X1,Y1) pontból az $y = Bx^2 + C$ parabolához huzott azon érintő R iránytangensét adja meg, amely esetén az (X2,Y2) érintési pontra: X2 < X1.

SUBROUTINE PAEGME1 (X1,Y1,R,A,E,X2,Y2,IER)
Legyen a P(X1,Y1) pont rajta az $y = Ax^2 + E$ parabolán. A P pont átmenő R iránytangensü egyenesnek a parabolával való másik (X2, Y2) metszéspontjának koordinátáit adja eredményül.

SUBROUTINE SZOGFEL (RM1, RM2, RM31, RM32)
Az RM1 és RM2 iránytangensü egyenesek két (egymásra merőleges) szögfelezőjének RM31, illetve RM32 iránytangensét adja meg.

SUBROUTINE PAEGME2 (X1,Y1,R,B,C,X2,Y2,X3,Y3,IER)
Az (X1,Y1) ponton átmenő, R iránytangensü egyenesnek az $y = Bx^2 + C$ parabolával való két metszéspontját (X2,Y2); (X3,Y3) adja meg (amennyiben azok valósak).

SUBROUTINE KOEGME1 (X1,Y1,R,XK,YK,RR,XM1,YM1,XM2,YM2,IER)
Az (X1,Y1) ponton átmenő, R iránytangensü egyenesnek az $y = Bx^2 + C$ középpontu, RR sugaru körrel való két metszéspontját (X2,Y2); (X2,Y3) adja meg (amennyiben azok valósak).

SUBROUTINE EGEGME (UX,UY,RQ,VX,VY,VQ,ZX,ZY)
Az (UX,UY) ponton átmenő, RQ iránytangensü és a (VX,VY) ponton átmenő, VQ iránytangensü egyenesek (ZX,ZY) metszéspontját adja meg.

SUBROUTINE EGEGME2 (B,RQ,VX,VY,VQ,ZX,ZY)
Az y = RQ*X+B és a (VX,VY) ponton átmenő, VQ iránytangensü egyenesek (ZX,ZY) metszéspontját adja meg.

SUBROUTINE PARR (XX1,YY1,XX2,XK,YK,EPX,IER,EPU,EPS,A,YY2)

Előállit egy (P1,P2) körivet (a kör középpontja XK,YK) oly módon, hogy P1(abszcisszája XX1) és P2(XX2,YY2) az $y=Ax^2-EPS$ parabolán (p_1) vannak, és k benne van a p_1 és az $y = Ax^2 + EPX$ (p_2) parabolák által meghatározott sávban. A p_1-től merőlegesen mért távolsága P_2-nek $\leq$ EPU.

SUBROUTINE PONTOS (XX1,YY1,XX2,XK,YK,IER,EPU,ELT,A,EP1,EP2, EPS,YY2)

A PARR-ban szereplő k körivet ugy állitja be, hogy p_1-nek p_2-től való (merőleges) távolsága $\leq$ EPS legyen.

SUBROUTINE ELEGER1 (X1,Y1,A,B,G,R,IER)

A P(X1,Y1) pontból az $(\frac{X}{A})^2 + (\frac{Y}{B})^2 = G$ ellipszishez huzott azon érintő R iránytangensét adja meg, amely esetén a (X2,Y2) érintési pontra: $X2 < X1$ ha $X1 \leq 0$, $Y1 < 0$.

SUBROUTINE ELEGME1 (X1,Y1,R,A,B,X2,Y2,E,IER)

Legyen a P(X1,Y1) pont rajta az $(\frac{X}{A})^2 + (\frac{Y}{B})^2 = E$ ellipszisen. A P ponton átmenő R iránytangensü egyenesnek az ellipszissel való másik (X2,Y2) metszéspontjának koordinátáit adja eredményül.

6. AZ IER HIBAPARAMÉTER LEHETSÉGES ÉRTÉKEI

IER=0 : sikeres futás

a/ parabolát közelitő szubrutinok esetén:

IER=1 a PAEGER1 helytelen bemenő adatok miatt nem ad érintési pontot

IER=2 a PAEGME1 nem ad valós metszéspontot

IER=3 a PAEGME2 nem ad valós metszéspontot

IER=5 a KOEGME1 nem ad valós metszéspontot

b/ az ellipszist közelitő szubrutinok esetén:

IER=1 az ELEGER1 helytelen bemenő adatok miatt nem ad érintési pontot

IER=2 az ELEGME1 nem ad valós metszéspontot

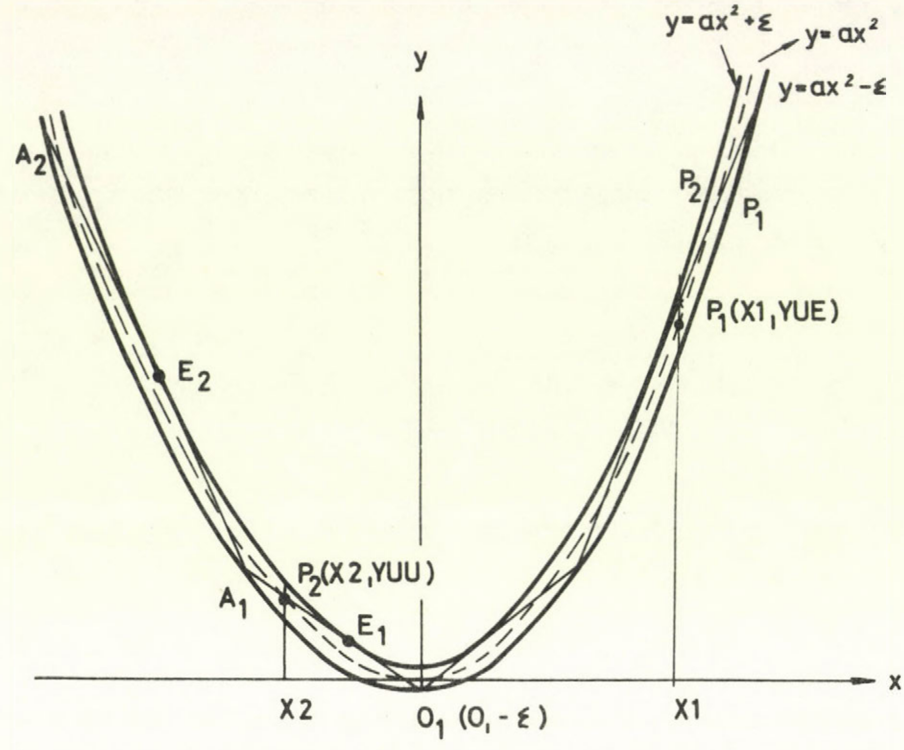
MEGJEGYZÉSEK:

1/ A szubrutinokat a könnyebb összehasonlithatóság kedvéért ugyanazokra a bemenő paraméterekre futtattuk le (a parabolák esetén).

2/ A közelitések során legfeljebb másodfoku, vagy másodfokura redukálható egyenletek megoldására kerül sor (a PAR2-öt kivéve).

3/ A dimenzionált változók (tömbök) indexhatárai a feladat nagyságának megfelelően változtathatók.

4/ Nem nehéz belátni, hogy a bemutatott módszerek az ábrán látható módon meg is valósithatók, megfelelő bemenő adatok esetén. Pl. a 3. ábrán a $\overline{P_2O}$ szögfelezőre merőleges és a parabolát E-ben érintő egyenes (m) $\overline{P_2P_1}$-el ill. $\overline{P_2P_3}$-al való M_1 vagy M_2 metszéspontja P_1 és P_2 ill. P_2 és P_3 között van. U.i. a $\overline{P_3P_2}$ és $\overline{P_2P_1}$ szakasz közül vegyük a rövidebbet (P_1P_2), és tekintsük a $P_2P_1P_1'$ Δ-et. A parabola nyilván metszi $\overline{P_1P_1'}$-öt P_1-ben és M-ben és a P_1M parabola iven van az E érintési pont; M_1 ill. M_2 igy csak P_1 és P_2 ill. P_2 és P_3 között lehet.

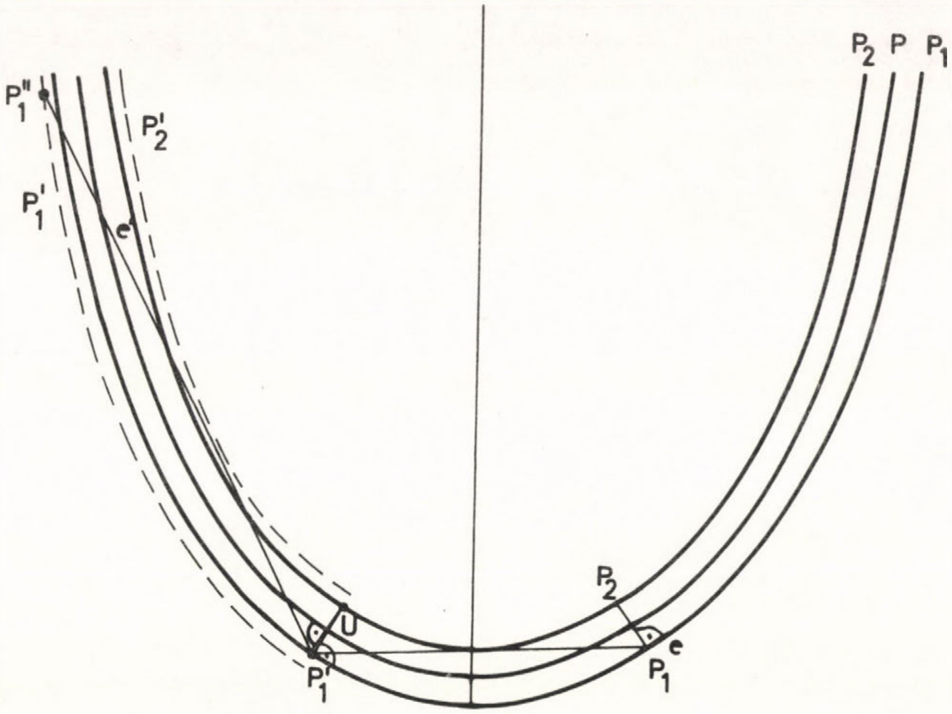


1. ábra

1/ Csak egyenesszakaszokkal való közelités.

0_1, A_1, A_2 az $y = ax^2 - \varepsilon$ parabolán (p_1) vannak, az $0A_1$ ill. A_1A_2 szakasz az $y = ax^2 + \varepsilon$ parabolát (p_2) az E_1 ill. E_2 pontokban érinti.

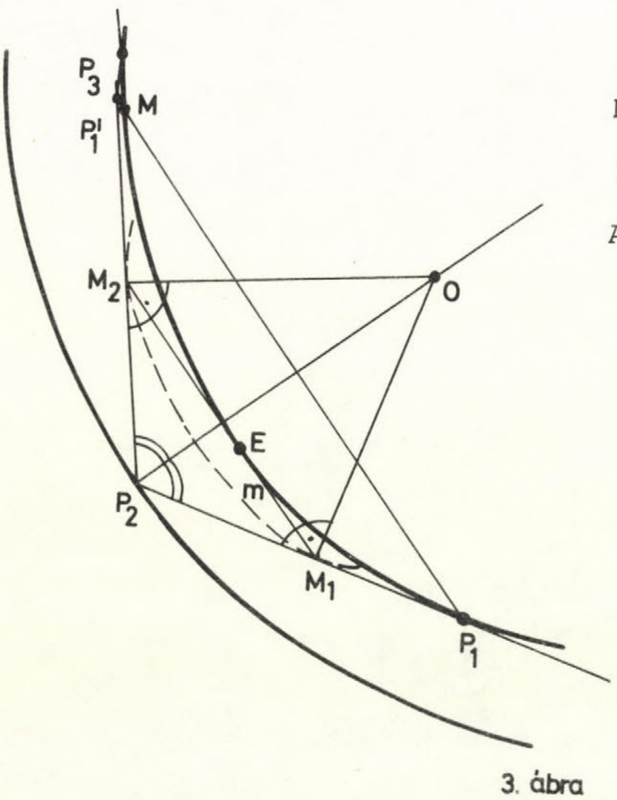
(egy adott pontból a p_2 parabolához huzott érintő meghatározása, majd ennek az érintőegyenesnek a p_1 parabolával való (megfelelő) metszéspontjának meghatározása másodfoku egyenletek megoldása, feltételek a diszkrimináns nullává válására)).



2. ábra

2/ a/ Egyenesszakaszok és körivek felváltva közelitenek.

Az 1/ esetnél a sarkok megfelelő lekerekitésével:



3. ábra

$OP_2P_3 = OP_2P_1$

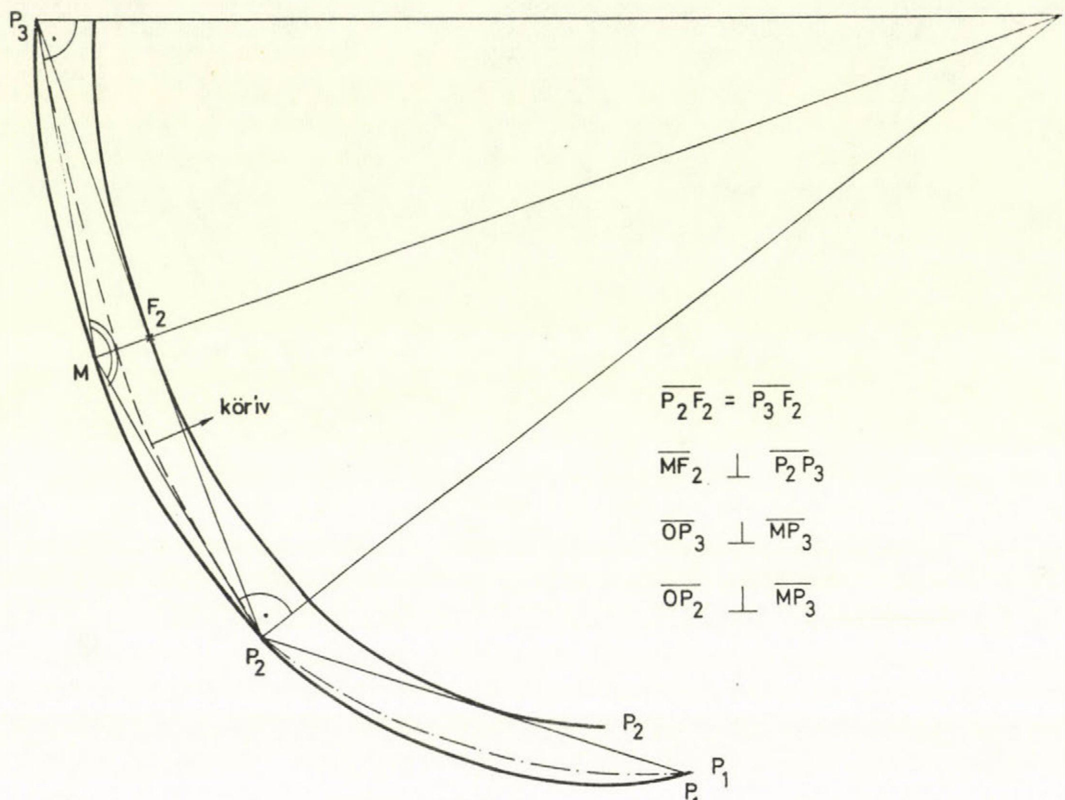
$M_1M_2 \perp OP_2$, M_1M_2 p-t E-ben érinti

$P_2M_2 \perp OM_2$, $P_1M_1 \perp OM_1$, $\overline{P_1P_2} = \overline{P_2P_1'}$

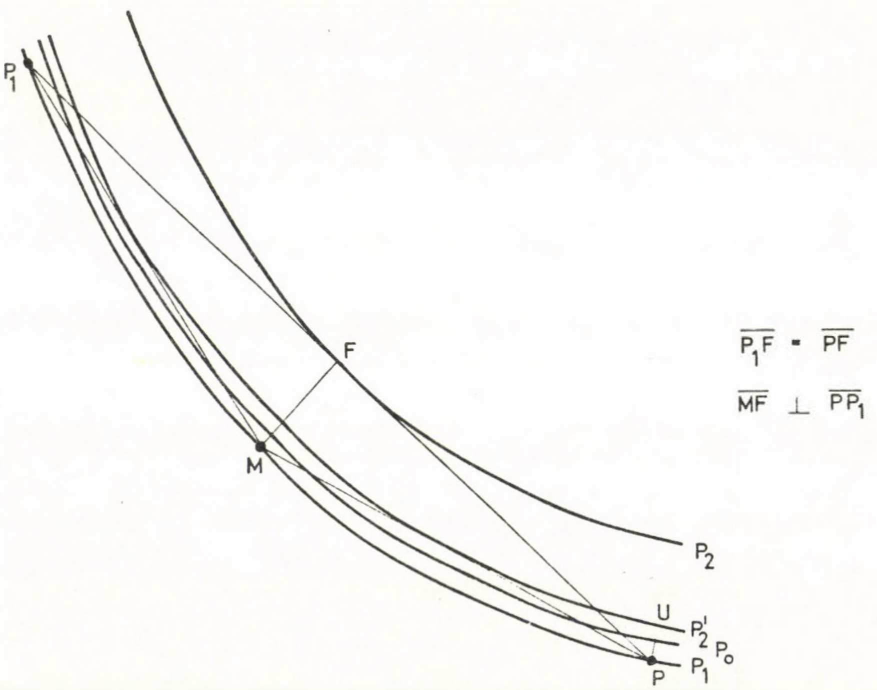
A kör középpontja O, sugara OM_1.

3/ Csak körivekkel közelitünk.

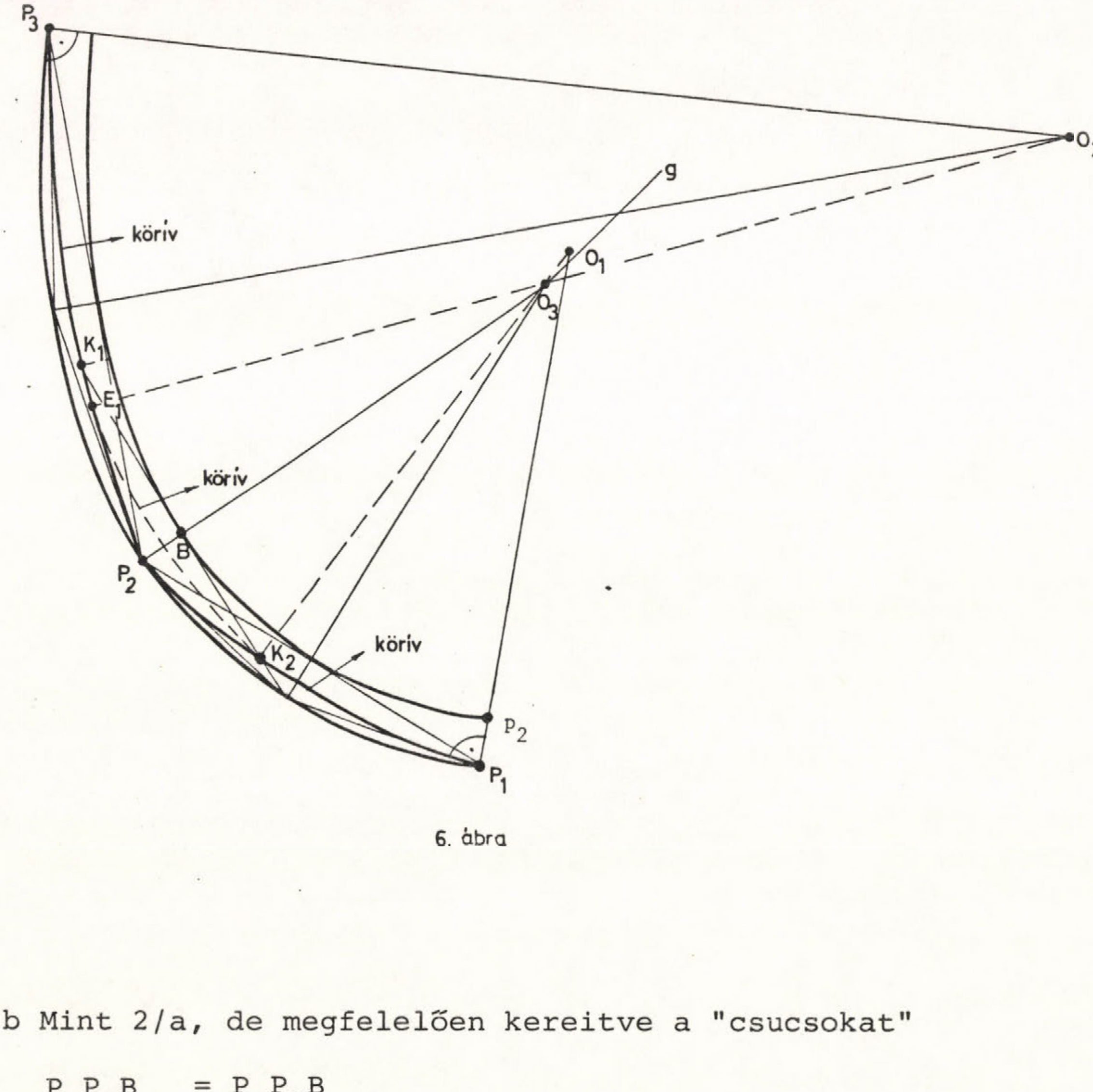
a/ az 1/ eset egyenesszakaszainak megfelelő helyettesitése körivekkel



4.ábra



5. ábra



6. ábra

3/b Mint 2/a, de megfelelően kereitve a "csucsokat"

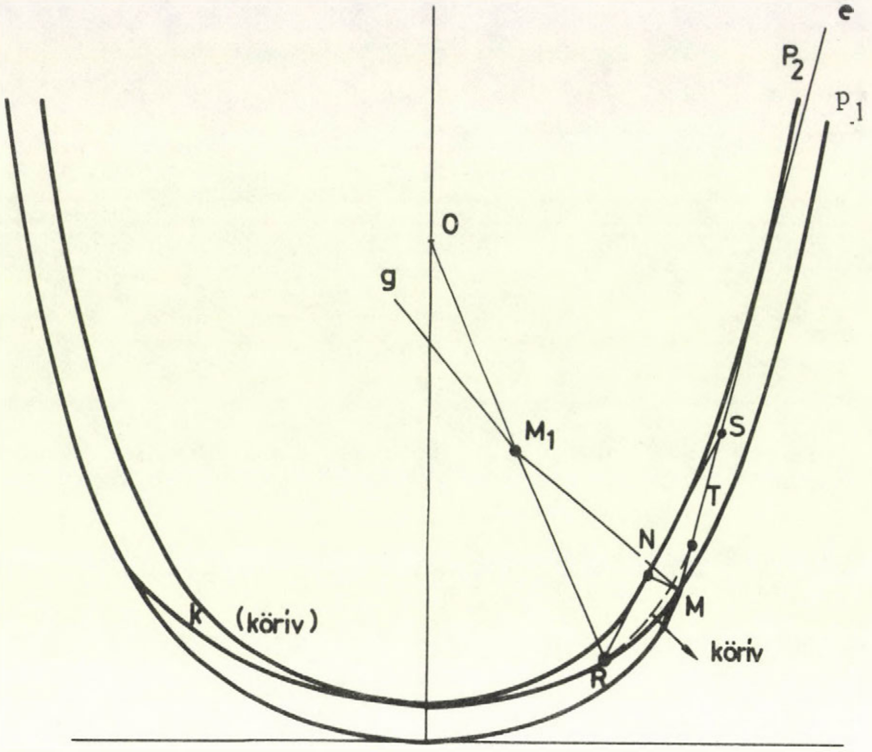
$P_3P_2B = P_1P_2B$

$\overline{BK_i} \perp \overline{BP_2}$

$\overline{BK_1}(\overline{BK_2})$ érinti p_2-t, $\overline{P_2K_2} < \overline{P_2K_1}$

A P_1P_2 és P_2P_3 körivtől egyenlő távolságra levő pontok mértani helye a g (legfeljebb) másodfoku görbe.

$\overline{K_2O_1}$ és g O_3 metszéspontja a kerekitő körív (K_2E_1) középpontja.



7. ábra

2/b Kezdetben (a parabola csucsához közel)
kör (k) közelités, utána egyenesszakaszok és körivek, mint 1/-ben, vagy 2/a/-ban.

k középpontja O

k és p_1 érintési pontja M

p_1 normálisa $\overline{MN}$

$\overline{MN} \perp \overline{NR}$, és $\overline{NR}$ érinti p_2-t

$\overline{NR}$ és k metszéspontja R

$\overline{NR}$ és e metszéspontja S

A k körivtől és az e egyenesszakasztól egyenlő távolságra levő pontok mértani helye a g görbe, $\overline{MR} < \overline{MS}$, OR és g metszéspontja (M_1) a kerekitő köriv ($\overline{RT}$) középpontja.

S u m m a r y

Approximation of parabola and ellipse by straight lines and arches

István Szepesvári

The task of the approximation of different curves by straight lines and arches appears in the computer - aided part programming for numerically controlled machine tools. In this paper we deal with this problem for parabola and ellipse. Starting from simple approximations we realize the approximation of given accuracy by more and more complicated FORTRAN-subroutines and less and less number of straight lines (arches).

Р Е З Ю М Е

Приближение параболы и эллипса с прямыми отрезками и дугами круга

И . Сепешвари

У станков с программным управлением возникает задача приближения различных кривых с прямыми отрезками и дугами круга. В этой статье занимаемся приближением параболы и эллипса. Начиная с простых приближений, приближение заданной точности дается при помощи все более сложных подпрограмм /составленных на языке FORTRAN/ и все меньше и меньше отрезков /дуг/.

MTA Számítástechnikai és Automatizálási Kutató Intézete, Közlemények 24/1980.

СИСТЕМА SIS79/GENERA

Пал Керекфи

1. *СИСТЕМА РЕДАКТИРОВАНИЯ ПРОГРАММ ГЕНЕРА*

1.1. Введение

Целью разработки программных средств обычно является создание процедур, способствующих облегчению и эффективности использования ЭВМ.

Генерирование /редактирование/ программ является методом помогающим программистам. Под генерированием подразумевается процедура, при помощи которой на базе команд /программ/, написанных на языке более высокого уровня, создается программа на языке более низкого уровня. /Этот язык имеет транслятор в данной программной конфигурации или встроенный в аппаратурные средства./ В таком смысле генерирующими программами являются трансляторы различных языков высокого уровня, а также предварительные трансляторы . Генерирование программ одновременно обеспечивает эффективность и более простое использование. Часто встречаем проблему, что программы, написанные в общей форме, недостаточно эффективные, и наоборот, программы, написанные с учетом эффективности, слишком сильно связаны с данной задачей и трудно их использовать даже при небольших изменениях параметров задачи. Поэтому в большинстве случаев дается компромисное решение, которое зависит от характера задачи, т.е. программа, которая более или менее общая и более или менее эффективная. В решении этих проблем поможет редактирование /генерирование/ программ.

Первой фазой проектирования системы является создание структуры эффективной целевой программы. Это является трудоемкой задачей и требует большое внимание. Написать процедуру генерирования уже значительно проще, особенно если она входит в систему, обеспечивающую подходящие условия для таких процедур. Система "Генера" была создана именно для такой цели, чтобы способствовать получению процедур редактирования.

1.2. Общее описание системы "Генера"

Целью системы "Генера" является включение процедур редактирования в единую систему. Упрощение и согласование работы программистов /системных программистов/, а также обеспечение единого, наглядного использования процедур компоновки. Система является модульной в большой степени и обеспечивает произвольное соединение процедур, и тем самым решение нескольких задач внутри одной, генерированной программы.

1.3. Команды

Произвольное соединение процедур редактирования в одну программу обеспечивается под управлением команд с произвольным содержанием /в нашем случае это программа на языке ФОРТРАН/. Команды находятся на входном файле. Строки, которые не содержат команд для "Генера", без изменения переносятся системой на выходной файл, а на место команд "Генера" записываются программные фрагменты, созданные процедурами редактирования. Команды управления начинаются специальным символом.

1.4. Построение системы "Генера"

Главными частями системы являются:

- Предварительная обработка
- Подготовка
- Обработка
- Окончание

Во время предварительной обработки программа производит формальную проверку команд и их параметров. В случае нахождения ошибки программа осведомляет об этом пользователя и после предварительной обработки программа останавливается. Во время проверки команд программа собирает данные о количестве памяти и файлов, необходимых для генерирования.

На основе этого в фазе <u>подготовки</u> проверяются необходимые для программы ресурсы.

Во вермя <u>обработки</u> программа перепишет содержание входного файла на выходной файл /генерированный программный файл/. Если при этом "Генера" находит команду /запись начинается с соответствующим специальным символом/, тогда прекратит переписывание и вызывает соответствующую подсистему для исполнения команды.

<u>После полной обработки</u> входного файла программа в соответствии с выбранным способом прогона обеспечивает трансляцию и прогон генерированной программы. Способ прогона /и другие параметры/ задаются пользователем с помощью команды OPTION.

Частью системы является подсистема <u>макрогенератора</u>. С помощью этого пользователь может использовать макро, определенные в системе, а также может определить новые.

2. *КОМАНДЫ В* SIS 79 */СИС 79/*

В дальнейшем изложены команды языка СИС 79, связанные с системой Генера. СИС 79 , как и другие системы построенные на Генера, легко расширяемый язык и может быть дополнен другими операциями, неперечисленными в дальнейшем. СИС 79 содержит процедуры, связанные со статистической обработкой данных. Процедуры сугубо математическо-статистического анализа не включены в СИС 79. Для решения таких проблем имеется много программ в распоряжении пользователей. /Благодаря гибким возможностям пользования системой Генера, эти процедуры без особых трудностей могут быть вызваны из программы./ При создании системы основной целью явилось обеспечение в легкообрабатываемой форме решения общих статистических задач и подготовительных операций математико-статистического анализа /контроля, кодирования, селекция, группирование, составление таблиц/. Надо отметить, что СИС 79 является не только языком для управления данными, но также обеспечивает эффективную обработку большой массы данных.

СИС 79 содержит задачи следующего типа:

1. Операции передачи и конверсии;
2. Топологическое отображение /контроль, кодирование/ и селекция;

3. Анализ статистических проб, создание основных данных для анализа математической статистики и составления таблиц;
4. Вывод форматов /таблиц/,
   Интерфейсы для статистических оценок и для передачи данных в другие системы;

В дальнейшем подробно представляем эти задачи.

2.1. Операции передачи и конверсии

Программа СИС 79 служит прежде всего для обработки большой массы данных, поэтому эффективность операций ввода и вывода играет в ней большую роль.

СИС 79 содержит четыре операции ввода и вывода:

LECTOR

SCRIPT

COBRED

COBWRT

2.1.1. Команда LECTOR

Задачей команды является считывание ввода и конверсия на двоичную форму. С командой /в настоящем варианте/ производится считывание положительных целых чисел.

Расположение данных входной записи задается пользователем в форматах языка Фортран. Структура записи определяется описанием, применяемым в КОБОЛ. Таким образом обеспечивается возможность совместного обращения с концентрированными данными. В отличии от КОБОЛ, здесь на каждом уровне могут задаваться форматы.

Целью команды является обеспечение быстрой конверсии. Это достигается эффективной процедурой, созданной для обработки ввода с форматами языка Фортран. В зависимости от структуры записи генерирования процедура ввода может даже на несколько порядков быстрее работать, чем традиционные процедуры в Фортране.

2.1.2. Команда SCRIPT

Задачей команды является печать данных, записанных в памяти, по форматам языка Фортран. Параметрическое изображение полностью совпадает с параметрическим изображением команды **LECTOR**. Первичной целью команды при этом является тоже ускорение конверсии.

2.1.3. Команда COBWRT

С использованием команды **COBWRT** данные могут быть переписаны на файл в уплотненной двоичной форме. Данные, записанные таким образом, могут быть считаны с файла с помощью команды **COBRED**.

При накоплении данных в формате символов потребность в памяти меньше чем при накоплении в двоичной форме /особенно в случае маленьких чисел/, но при этом исполнение расчета замедляется из-за конверсии. Процедура **COBWRT** обеспечивает запоминание неотрицательных целых чисел в уплотненной и в то же время в двоичной форме, необходимой для быстрой обработки. В каждом машинном слове помещается несколько данных. Каждый из данных занимает столько разрядов, сколько необходимо для запоминания его максимального значения.

При пользовании этой процедурой пользователь кроме названия печатаемых данных задает также ограничение по величине данного /или длину в разрядах/. На основе этого процедура распределит данные в уплотненной форме.

2.1.4. Команда COBRED

Командой производится считывание уплотненной двоичной записи, произведенной командой COBWRT. Пользователь задает описание такой же записи, как в команде COBWRT и на основе этого процедура производит первоначальные данные.

2.2. Операции топологического отображения и селекции

Эта группа операций содержит все задачи подготовки данных, требующие операции отображения /кодирование, анализ функций/. Такими являются простые и сложные задачи проверки данных, операции селекции и принятия решений. Эти задачи исполняются следующими командами:

BOUNDS

GRAPH

DECISE

При каждой операции отображения необходимо каким-то образом описать процедуру. Это не сложно, если речь идет о функции, заданной формулой, но в случае сложных функций, которые могут быть описаны только при помощи таблицы значений это может привести к таблицам грамоздкого размера. Описание, проверка и запоминание таких таблиц очень сложная проблема. В системе СИС 79 эта проблема решается при помощи специфичной системы запоминания и подпрограммы "Мешок".

2.2.1. Подпрограмма "Мешок"

Функцией подпрограммы является реализация процедур для удобного заполнения таблиц /цифровой перечень слов, таблица переходов, таблицы значений функций/ в центральной памяти. Применение программы обеспечивает следующие выгоды:

1. Пользователь может описать таблицы значений в самой удобной, наглядной и плотной форме.
2. Описание таблиц значений синтактически проверяется подпрограммой и дается подробный сигнал распозновения ошибок.
3. Обеспечивается быстрое и удобное отображение некоторых многопеременных функций.

2.2.2. Команда BOUNDS

С помощью этой команды проверяется, что значение переменных не выходит ли из заданных ограничейний. Это может являться начальным шагом контроля данных и подготовкой к применению нижеописываемой команды **GRAPH**. Генерированная программа информирует пользователя о погрешностях в данных при помощи переменного счетчика ошибок и массива сбора кодов проверки.

2.2.3. Команда GRAPH

С помощью этой команды может быть создан набор прог-программ для оценки таблиц перекодирования и провер-ки. Программа проверяет данные на основе таблицы, заполненной с помощью подпрограммы Мешок, а также на-ходит ошибочные комбинации значений и определяет значение функции, относящейся к безошибочным комби-нациям. С использованием этой команды заодно можно заполнить таблицу, так как она непосредственно свя-зана с подпрограммой Мешок. При прогоне генерирован-ной программы /также как в случае команды BOUNDS/ пользователь получит информацию о данных с погреш-ностями при помощи счетчика ошибок и массива сбора кодов проверки.

Таблица проверки и перекодирования служит для описа-ния целочисленных функций и применяется значительно эффективнее, чем традиционные таблицы значений, если в таблице значений некоторые части повторяются.

Функция описывается с помощью иерархического графа, размещенного на уровнях по отдельным переменным: на первом уровне решение принимается в зависимости от значения первой переменной, на втором уров-не - от значения второй переменной и т.д. Каждый уровень включает в себя один или несколько частичных таблиц /мешков/, содержащих информацию о той части возможных значений переменной соответствующего уровня, которая попадет в один интервал. При этом получена информация о том, что функция существует ли на данное значение, какого её значение, а также получена команда по дальнейшему анализу.

2.2.4. Команда DECISE

При статистической обработке вместо интегрирования системы данных часто целесообразно прибегать к разделению на подсистемы по различным информационным потребностям. Аспекты разделения часто описываются с весьма сложными логическими выражениями. Команда **DECISE** облегчает описание сложных логических выражений и осуществляет эффективную оценку логических выражений. Логические функции, описанные таким образом могут применяться не только при селекции статистических множеств данных, но также при других задачах принятия решений.

2.3. Анализ статистических проб

При статистическом анализе требуются данные не только об индивидумах, но также о характере групп. Взятие пробы производится по индивидумам, поэтому при статистическом анализе одним из самых важных и характерных шагов является группировка и суммирование данных, определение частоты случаев. Это служит основой для общего анализа и для математической статистики.

В СИС 79 могут быть собраны частоты, а также сумма значений одной и более выбранных переменных /например, суммы произведений и квадратов для составления корреляционной матрицы/.

В системе обеспечивается обработка большой массы данных при помощи быстрой процедуры сбора.

2.4. Составление таблиц

В СИС 79 составление статистических таблиц производится на основе заранее собранных частот и сумм, поэтому для составления каждой таблицы /в зависимости от размера таблицы/ требуется только несколько секунд машинного времени.

SUMMARY

SYSTEM SIS79/GENERA

P. Kerékfy

Generating programs is a well-known method for easy and efficient programming. Our system SIS79 is based on generator system GENERA. GENERA integrates any number of generator subsystems into a single precompiler. A file of host language statements and directives is processed to create the host language program.

Statistical information system SIS79 contains a number of data handling and data modification tasks. Interfaces to statistical packages can be established.

A SIS79/GENERA rendszer

Kerékfy Pál

A programok generálása egy jól ismert módszer a programozásban. A SIS79 rendszert a GENERA generáló rendszer segitségével épitettük fel. A GENERA tetszőleges számu generáló eljárást egyesit egyetlen előforditóvá. A rendszer alapnyelven irt forrásprogramban elhelyezett utasitások alapján hozza létre a kivánt programot.

A SIS79 statisztikai információs rendszer adatkezelő, konverziós és függvényeket kiértékelő eljárásokat tartalmaz. A különböző statisztikai programcsomagokhoz könnyen kapcsolható.