G. DIACONESCU
Department of Economic Cybernetics
Academy of Economic Studies

The SA-V Algorithm for Retrieval after Segments of Values within an Attribute

The study tackles a variant of the method of inverting the files aiming at formulating an algorithm for retrieval after value intervals. The algorithm is presented in a most general form and is based on the notion of SA-V pair (subattribute-value).

The algorithm is described in the form of a booleen expression in which the terms are bit patterns of the SA-V record and the operations are the sums and logical products generated by the ends of the value interval.

An iterative form of the algorithm is outlined in which the so-called stellated sums appear. The use of the advocated organization and of the SA-V algorithm is recommended for data bases with a low volatility and at which the retrieval demands after value intervals are frequent.

1. INTRODUCTION

The information systems based on the electronic computer are of quite recent date in data processing. Nevertheless, the hardware and software systems functioning at present allow some large-scale achievements in solving the efficient information problems regarding enterprise administration, population, documentation, invention patents, criminal and juridical data, economic data on planning, technological data, a.s.o. The general tendency is to approach such a system within the conception of the data base, a conception in which an important role is assigned to the retrieval function. In this way, the retrieval algorithm plays an essential role in the context of the information systems based on the electronic computer.

The two main categories of problems implied in the process of information retrieval regard:

1) either the *identification* of an item of data (or of a structure of items) through identity attributes from a set of items (the set can be structured). The retrieval consists in locating the information item given being its identity.

2) or the *seeking* of a subset through seeking attributes from a set of information items. The retrieval consists in determining all items which have the value of the seeking attribute in common.

The identification type retrieval is strictly necessary in the electronic data processing, if one wishes the direct access to data. A number of techniques which have developed are known in this connection, for example, the HASH ADDRESSING algorithm, etc. The systems which are mainly achieved for retrieval by identification have lately been called operation systems (for example IMS-IBM).

The problem of finding a subpopulation of individuals in a population, based on common values of some attributes (sex, profession, foreign languages known, etc.) is typical of the second way of approach (retrieval by seeking).

In the terms of reference information, the population becomes the set of books or items, the subpopulation is similar to the bibliographical list, etc. The systems which are based mainly on retrieval by seeking are called executive systems (for example GIS-IBM).

In more general terms, the retrieval by seeking implies:

— a file made up of logical record (items), each logical record having a number of fields (attributes);

— the result of the seeking is a list of all logical records which have the values of the seeking attributes in common.

This last way of approach will be adopted in the following and developed for describing an algorithm which allows retrieval according to seeking attributes, for intervals of values. The method introduces the notion of subattribute-value (SA-V) and is based on the technique of inverted files. The formulation of this algorithm is essentially based on the notion of subattribute-value.

The retrieval according to value intervals within an attribute is a function implemented in a number of systems (GIS-IBM, SOCRATE-IRIS, etc.), the methods used in the implementation having not been published, however.

2. NOTATIONS AND DEFINITIONS

An attribute is the name of a field of the logical record. For a file of personell, in a logical record we can find the attributes "sex", "wage", etc.

A value is the content of a field in a certain recording. For example, in a certain record, the attribute "wage" may have the value 2400, in another, 2 900, etc.

The pair A-V is the attribute-value pair.

The recording thus appears as a collection of A-V pairs.

We may distinguish:

a) Attributes covering distinct values from a record to another in the entire file. These attributes may be called identity attributes. An A-V pair in this case represents the identity of a record.

b) A-V pairs common to several records. The attributes whose values can be repeated from a record to another will be called seeking attributes, for example "wage", "age", etc.

3. THE ROLE OF THE IDENTITY ATTRIBUTES AND OF THE BIT PATTERNS OF THE INVERTED RECORD

The technique of inverted files for solving the problems of a seeking type retrieval is well known; it implies: a) the existence of an object file (a file in which one seeks) containing the object records, each having an identity in the meaning mentioned above; b) the existence of an inverted file (with the aid of which one seeks in the object file), containing the inverted records. The identity of an inverted record is given by one of the A-V pairs of the object file, A being a seeking attribute. The content of an inverted record is the list of all identities of the object records including the A-V pair which identifies the respective inverted record.

Of great use evinces sometimes the replacement of the list of identities in the inverted record (a list which varies in length, in principle), with a sequence of bits of invariable length, equal to the number of object records, called the bit pattern of inverted record. This is an incidence vector in which the bit corresponding to a certain object record has the value 1 or 0, according to the membership or non-membership to the object record of the A-V pair identifying the inverted record.

4. PUNCTUAL SEEKINGS, MULTIPUNCTUAL SEEKINGS, AFTER A-V SEGMENTS.

With the aid of inverted records, seeking is made in three steps, namely:

a) We have the object file and the inverted file. An A-V pair is presented as a seeking argument. Accordingly, by an identifying retrieval, the inverted record having the identity equal to the seeking argument is located; thus said, the access is achieved to a corresponding inverted record.

Seeking which implies access to a single inverted record will be called punctual seeking. Most often, the seeking argument is complex and is made up of a number of A-V pairs linked between them by logical operators. The seeking implying access to several inverted records will be called multipunctual seeking. An example of multipunctual seeking: the seeking argument is (FUNCTION= ING. OR. FUNCTION = TECHN). AND (NO. CHILDREN = ø). In this example, the multipunctual seeking is made up of three punctual seekings, linked with the logical operators. OR and. AND.

b) In the second stage, the bit patterns of all inverted records of a multipunctual seeking are logically operated according to the structure of the complex seeking argument. The result of the second stage is the bit patterns of the answer to the complex (or simple) seeking argument.

c) In the third stage, the output is produced. The bit patterns of the answer is used either for an identifying retrieval in the object file with the

adequate listing of the recordings or for the simple counting of the corresponding records, etc.

Of course, problems arise in using the bit patterns in connection with the representation of the one to one correspondence between the bits and the object records, but we shall not detail this aspect.

On the contrary, we shall outline that sometimes it is very important that the seeking argument be not a simple A-V pair but an A-V segment, namely an attribute followed not by a value but by a continuous range of values (numerical); for example: the list of all employees with the salary within the limits WAGE = (1500—2800) is requested.

Obviously, the seeking according to an A-V segment may, in some cases, be achieved as a multipunctual seeking. For example: the seeking according to the argument WAGE = (1200—1205) can be achieved by a multipunctual seeking; for example: WAGE = 1200. OR. WAGE = 1201. OR. WAGE= 1202.OR.WAGE = 1203. OR.WAGE = 1204.OR. WAGE = 1205.

If, however, the range of values is very large, practical considerations of implementation considerably reduce the possibility of seeking according to an *A-V* segment, by multipunctual *A-V* seekings.

5. THE NOTION OF SUBATTRIBUTE AND ITS LINK WITH THE SEEKING ACCORDING TO A SEGMENT OF VALUES WITHIN AN ATTRIBUTE

To clarify the following notions, we shall consider the case of seeking according to a segment of values within the attribute "wage". Thus, the question is about a personell-object file and about an inverted file in which an inverted record exists for each value of the salary. Since, considerations of implementing the inverted file (deletions, additions, re-organizations, etc.) imply the taking into consideration of all possible values of the wage for an enterprise employees, approximately 9000 inverted records result, with the identities:

WAGE = 0800
WAGE = 0804
. .
. .
. .

WAGE = 9999

Admitting now that the list of all employees with the salary comprised between 5000 and 9999 is requested, it results that we can obtain the answer by a multipunctual seeking with 5000 points, namely: WAGE = 5000. OR. WAGE = 5001.OR ... OR.WAGE = 9999.

It can be seen how unpractical is this way of approaching the problem: a big amount of storage is used with the inverted records but the most important thing, the second stage of the previously defined seeking, lasts very much (logically 5000 BIT PATTERNS are operated), Obviously, such a seeking may bring less advantages in comparison with the purely sequential access covering the object file.

Another way of approach consists in the following:

I. Let us consider a pair A-V. We admit that the value occupies an invariable number of characters from a pair to another. We shall assign a name to each character of the value in the pair A-V. In this way, we can see that we may consider the pair A-V: WAGE = 1425 with a logical record made up of the pairs A-V.

figure — thousands — wage = 1
figure — hundreds — wage = 4
figure — scores — wage = 2
figure — units — wage = 5

In this way, the name of SUBATTRIBUTE for the name of a character of the value in a pair A-V is justified.

We can thus say that a pair A-V is a couple made up of a number of subattribute -value pairs (SA-V). The pair A-V: WAGE = 1425 is thus made up of the pairs SA-V:

figure-thousands-wage = 1
figure — hundreds — wage = 4 pairs SA-V
figure — scores — wage = 2
figure — units — wage = 5

II. In the inverted file we can assign an inverted record not to each A-V in the object file, but, rather to each pair SA-V.Thus, a considerable reduction of the number of inverted records results. Under these conditions, instead of the 9000 inverted records necessary to the pairs A-V generated by the attribute wage, we shall have:

— 4 subattributes

— 40 pairs SA-V, thus 40 inverted records SA-V (each subattribute may cover the values 0. 1,..., 9).

III. With this organization, the punctual seeking becomes complicated a little, artificially becoming a multipunctual seeking. For example, the punctual seeking WAGE = 1871 becomes a multipunctuql seeking:
(figure units wage = 1), AND. (figure scores wage = 7). AND.
(figure hundreds wage = 8). AND. (figure thousands — wage = 4). Obviously, in the implementation of the programs, we can use the reformulation of a punctual seeking A-V in the multipunctual seeking SA-V so that the user should not be implied. For this, the seeking demand in the natural form (A-V) will be introduced.

The punctual seeking A-V by multipunctual seekings SA-V may be useful, for example, in the inverted file of names. In this case, any name of 10 characters, for instance, can be an argument of seeking if we have 270 inverted records SA-V (27 for each subattribute representing the number of letters in the alphabet). The seeking A-V for the argument NAME=ION would become the multipunctual seeking SA-V:

first — letter — name = I. AND.
second — letter — name = O. AND.
third — letter — name = N.

IV. *Seeking after segments A-V* appears with the notion of a subattribute also as a multipunctual seeking but more efficient than in the case the points are A-V.

If we define as length of seeking the number of inverted records necessary to obtain the answer it is clear that the length of the seeking in representing the inverted record on SA-V is smaller than the length of seeking in representing the inverted record on A-V the difference becoming sensible in the case of long segments. The formulation of the SA-V algorithm thus appears in an extremely general way.

The seeking after a segment A-V is a logical expression in which the terms are punctual seekings generated by the ends of the segment. It remains to be inferred effectively: which are the punctual seekings SA-V generated by the ends of a segment A-V ; 2) what is the logical expression linking the punctual seekings SA-V.

6. THE FORMULA OF THE SA-V ALGORITHM

The formula of the algorithm is obtained by the reduction to punctual seekings and the computation of the logical expression. For this, we shall proceed by induction. Consider the segment $a_4a_3a_2a_1 - b_4b_3b_2b_1$. Thus, we have 4 subattributes (characters of the value).

We mean by a_i, b_i figures in the positional representation of the values of the ends of the segment. Usually, this representation is the representation in the usual 10 radix, but, for generalization we may consider any base. Moreover, we shall consider that the maximum value of the subattribute 1 is different from the maximum value SA_2 etc, and we shall note with V_1, V_2, ..., V_4 the maximum values of the subattributes.

We may consider:

$a_4 \leqslant b_4$,the general case (in which the interval may start with pairs (a_i, b_i) with equal components) being dwelt with in the second stage of induction

$$\left.\begin{array}{l} 0 \leqslant a_1,\ b_1 \leqslant V_1 \\ 0 \leqslant a_2,\ b_2 \leqslant V_2 \\ 0 \leqslant a_3,\ b_3 \leqslant V_3 \\ 0 \leqslant a_4,\ b_4 \leqslant V_4 \end{array}\right\} \quad \begin{array}{l} \text{(usually:} \\ V_1 = V_2 = V_3 = V_4 = 9) \end{array}$$

In the first stage, we shall try to decompose the initial segment into disjunctive segments so that each of them be characterized by a single punctual seeking, according to subattribute 4.

Thus: $a_4 \quad a_3 \quad a_2 \quad a_1 - b_4 \quad b_3 \quad b_2 \quad b_1$

is decomposed into:

m) $a_4\ a_3\ a_2\ a_1 - \quad a_4\ V_3\ V_2\ V_1$

$a_4+1\ 0\ 0\ 0 \quad -a_4+1\ V_3\ V_2\ V_1$

$a_4+2\ 0\ 0\ 0 \quad -a_4+2\ V_3\ V_2\ V_1$

.

.

.

$b_4-1\ 0\ 0\ 0 \quad -b_4-1\ V_3\ V_2\ V_1$

n) $b_4\ 0\ 0\ 0 - \quad b_4\ b_3\ b_2\ b_1$

It can be seen immediately that all segments except for m and n may be sought with the single punctual seeking SA-V; of the type of the subattribute $4 = a_4 + i$.

We shall note this seeking with $R_4^{a_4+i}$. We can note for homogeneity the seeking after a segment also with R, the segment being presented below as an index:

$$R_{a_4\ a_3\ a_2\ a_1 - b_4\ b_3\ b_2\ b_1}$$

Notations $R_4^{a_4+i}$ and $R_{\blacksquare\blacksquare\blacksquare\blacksquare - \blacksquare\blacksquare\blacksquare\blacksquare}$ may be considered as being the BIT PATTERNS of the respective answers. With this last notation we can define the logical operations of logical sum a logical product. We shall use the multiplication sign for the logical product and the addition sign for the logical sum.

We thus have:

$$R_{(a_4+1)\ 0\ 0\ 0 - (a_4+1)\ V_3\ V_2\ V_1} = R_4^{a_4+1}$$

$$R_{(a_4+2)\ 0\ 0\ 0 - (a_4+2)\ V_3\ V_2\ V_1} = R_4^{a_4+2}$$

$$\vdots$$

$$R_{(b_4-1)\ 0\ 0\ 0 - (b_4-1)\ V_3\ V_2\ V_1} = R_4^{b_4-1}$$

and correspondingly:

$$R_{a_4\ a_3\ a_2\ a_1 - b_4\ b_3\ b_2\ b_1} =$$

$$= R_{a_4\ a_3\ a_2\ a_1 - a_4\ V_3\ V_2\ V_1} + R_4^{a_4+1} + R_4^{a_4+2} + \ldots + R_4^{b_4-1} +$$

$$+ R_{b_4\ 000 - b_4\ b_3\ b_2\ b_1}.$$

The method is further applied: the segments

$$a_4\ a_3\ a_2\ a_1 - a_4\ V_3\ V_2\ V_1$$

and

$$b_4\ 0\ 0\ 0 - b_4\ b_3\ b_2\ b_1$$

are decomposed into disjunctive segments so that each of them be characterized by punctual seekings;

$$a_4\ a_3\ a_2\ a_1 - a_4\ V_3\ V_2\ V_1$$

is decomposed into:

a_4	a_3	a_2	$a_1 - a_4$	a_3	V_2	V_1
a_4	$a_3 + 1$	0	$0 - a_4$	$a_3 + 1$	V_2	V_1
a_4	$a_3 + 2$	0	$0 - a_4$	$a_3 + 2$	V_2	V_1
$\vdots$						
a_4	V_3	0	$0 - a_4$	V_3	V_2	V_1

Similar to the above method we have:

$$R_{a_4\,a_3\,a_2\,a_1-a_4\,V_3\,V_2\,V_1} = R_{a_4\,a_3\,a_2\,a_1-a_4\,a_3\,V_2\,V_1} + \\ + R_4^{a_4} \cdot R_3^{a_3+1} + R_4^{a_4} \cdot R_3^{a_3+2} + \ldots + R_4^{a_4} \cdot R_3^{V_3}.$$

Correspondingly for:

$$b_4\ 0\ 0\ 0 - b_4\ b_3\ b_2\ b_1$$

we have:

$$R_{b_4\,0\,0\,0-b_4\,b_3\,b_2\,b_1} = R_4^{b_4} \cdot R_3^0 + \\ + R_4^{b_4} \cdot R_3^1 + \ldots + R_4^{b_4} \cdot R_3^{b_3-1} + R_{b_4\,b_3\,0\,0-b_4\,b_3\,b_2\,b_1}.$$

The initial seeking is written now by taking into consideration also the associativeness of the logical process:

$$R_{a_4\,a_3\,a_2\,a_1-b_4\,b_3\,b_2\,b_1} = R_{a_4\,a_3\,a_2\,a_1-a_4\,a_3\,V_2\,V_1} + \\ + R_4^{a_4}(R_3^{a_3+1} + R_3^{a_3+2} + \ldots + R_3^{V_3}) + \\ + R_4^{b_4}(R_3^1 + R_3^2 + \ldots + R_3^{b_3-1}) + R_{b_4\,b_3\,00-b_4\,b_3\,b_2\,b_1}.$$

We still have to decompose into punctual seekings the segments

$$R_{a_4\,a_3\,a_2\,a_1-a_4\,a_3\,V_2\,V_1}$$

and

$$R_{b_4\,b_3\,0\,0-b_4\,b_3\,b_2\,b_1}.$$

We can write:

$$R_{a_4\,a_3\,a_2\,a_1-a_4\,a_3\,V_2\,V_1} = R_4^{a_4} \cdot R_3^{a_3} \cdot R_2^{a_2}(R_1^{a_1} + \ldots + R_1^{V_1}) + \\ + R_4^{a_4} \cdot R_3^{a_3} \cdot R_2^{a_2+1} + \ldots + R_4^{a_4} R_3^{a_3} R_2^{V_2}$$

and similarly

$$R_{b_4\,b_3\,0\,0-b_4\,b_3\,b_2\,b_1} = R_4^{b_4} \cdot R_3^{b_3} \cdot R_2^0 + R_4^{b_4} \cdot R_3^{b_3} R_2^1 + \ldots + \\ + R_4^{b_4} \cdot R_3^{b_3} \cdot R_2^{b_2}(R_1^0 + R_1^1 + R_1^2 + \ldots + R_1^{b_1}).$$

The initial seeking becomes after all operations are made:

$$R_{a_4\,a_3\,a_2\,a_1-b_4\,b_3\,b_2\,b_1} = \\ = R_4^{a_4}(R_3^{a_3}(R_2^{a_2}(R_1^{a_1} + (R_1^{a_1+1} + \ldots + R_1^{V_1}) + R_2^{a_2+1} + \ldots + \\ + R_2^{V_2}) + R_3^{a_3+1} + \ldots + R_3^{V_3}) + R_4^{a_4+1} + R_4^{a_4+2} + \ldots + \\ + R_4^{b_4-1} + R_4^{b_4}(R_3^{b_3}(R_2^{b_2}(R_1^{b_1} + (R_1^{b_1-1} + \ldots + R_1^0)) + \\ + R_2^{b_2-1} + R_2^{b_2-2} + \ldots + R_2^0) + R_3^{b_3-1} + R_3^{b_3-2} + \ldots + R_3^0).$$

Analysing the above formula we shall seek the aspect of the formula in the most general form. We shall thus consider the segment in the form:

$$X_m X_{m-1} \ldots X_1 a_n a_{n-1} \ldots a_1 - Y_m Y_{m-1} \ldots Y_1 b_n b_{n-1} \ldots b_1$$

the figures noted with X in m number, being equal to figures Y respectively and a_n and b_n being the first pair of different figures.

The seeking of the segment may be expressed by:

$$R_m^{X_m} \cdot R_{m-1}^{X_{m-1}} \cdot R_{m-2}^{X_{m-2}} \dots R_1^{X_1} \cdot R_{a_n a_{n-1} \dots a_1 - b_n b_{n-1} \dots b_1}$$

It still remains to value:

$$R_{a_n a_{n-1} \dots a_1 - b_n b_{n-1} \dots b_1}.$$

Obviously, in the case of the identical ends of segments (interval) the seeking is expressed entirely as $\prod_{i=1}^{m} R_i^{X_i}$.

Introducing the recurrence formulas for defining the STAR sum:

$$\sum_{i=0}^{b_k}{}^{*} R_k^i = R_k^{b_k} \cdot \sum_{i=0}^{b_{k-1}}{}^{*} R_{k-1}^i + \sum_{i=0}^{b_k - 1} R_k^i \tag{1}$$

$$\sum_{i=a_k}^{V_k}{}^{*} R_k^i = R_k^{a_k} \cdot \sum_{i=a_{k-1}}^{V_{k-1}}{}^{*} R_{k-1}^i + \sum_{i=a_{k+1}}^{V_k} R_k^i \tag{2}$$

with

$$\sum_{i=0}^{b_1}{}^{*} R_1^i = \sum_{i=0}^{b_1} R_1^i \text{ and } \sum_{i=a_1}^{V_1}{}^{*} R_1^i = \sum_{i=a_1}^{V_1} R_1^i \tag{2 bis}$$

we have:

$$R_{a_n a_{n-1} \dots a_1 - b_n b_{n-1} \dots b_1} = \sum_{i=a_n+1}^{b_n - 1} R_n^i + R_n^{a_n} \left(\sum_{i=a_{n-1}}^{V_{n-1}}{}^{*} R_{n-1}^i \right) + R_n^{b_n} \cdot \sum_{i=0}^{b_n - 1}{}^{*} R_{n-1}^i,$$

and

$$R_{X_m X_{n-1} \dots X_1 a_n \dots a_1 - Y_m Y_{m-1} \dots Y_1 b_n \dots b_1} =$$

$$= \prod_{i=1}^{m} R_i^{X_i} \cdot \left[\sum_{i=a_n+1}^{b_n - 1} R_n^i + R_n^{a_n} \cdot \left(\sum_{i=a_{n-1}}^{V_{n-1}}{}^{*} R_{n-1}^i \right) + R_n^{b_n} \cdot \left(\sum_{i=0}^{b_{n-1}}{}^{*} R_{n-1}^i \right) \right] \tag{3}$$

7. CONCLUSIONS

1. Through formula 3 a seeking algorithm after intervals of values within an attribute was outlined, the interval being defined in the most general form. The algorithm is based on the existence of inverted records SA-V.

2. The organization of inverted files according to method SA-V allows:

a) the efficient retrieval after intervals of values;

b) the reduction of the number of inverted records as compared to the inversion only according to the attribute.

Supposing that the object records have a fixed structure it results that the number of inverted records SA-V is equal to $\sum_{i=1}^{n} V_i$ where V_i = the maxi-

mum number of values of each subattribute (a character in the recording) and n is the number of subattributes (characters).

c) reduced necessities for reorganizing the inverted file.

At the same time, the SA-V organization of the inverted file implies a considerable effort for updating. This may become a major disadvantage in the event of an application in which the time for answer to questions in the data base is less important and in case of a great volatility of the data base.

The SA-V organization is thus indicated in less volatile data base, with frequent demands with regard to intervals of values and to which the rapidity with which the answer is received is very important.

REFERENCES

BURTON H. BLOOM: *Some Technique and Trade-offs Affecting Large Data Base Retrieval Times.* JACM, 1971.

* * * : *Socrate* (documentation for use), CII, France, 1973.

* * * : *Mistral* (documentation for use), CII, France, 1972.

* * * : *Generalized Information System* — IBM, 1972.

ALGORITHME SA-V POUR RETROUVER L'INFORMATION SELON LES INTERVALLES VALORIQUES

L'auteur s'occupe d'une variante de la méthode d'inversion des fichiers ayant pour but de formuler un algorithme pour retrouver l'information selon les intervalles valoriques. L'algorithme est présenté sous la forme la plus générale et a pour base la notion de paire SA-V (sous-attribut-valeur). L'algorithme est présenté sous la forme d'une expression booléenne où les termes sont les modèles en bits des enregistrements SA-V et les opérations sont les sommes et les produits logiques engendrés par les extrémités de l'intervalle valorique.

On met en évidence une forme itérative de l'algorithme où apparaissent les sommes à astérisque.

L'utilisation de l'organisation prévue et de l'algorithme SA-V s'avère utile pour la base de données à volatilité faible et où il existe de fréquentes demandes pour retrouver l'information selon les intervalles valoriques.

АЛГОРИТМ SA-V ДЛЯ НАХОЖДЕНИЯ ПО СЕГМЕНТАМ ВЕЛИЧИН В РАМКАХ АТРИБУТА

Настоящая статья рассматривает вариант метода реверса картотек с целью формулирования алгоритма для нахождения по интервалам величин. Алгоритм представлен в наиболее общем виде и имеет в своей основе понятие пары SA-V (субатрибут — величина). Алгоритм является открытым в виде выражения Бооле, в котором члены представляют собой модели записей SA-V в битах, а операции являются логическими суммами и произведениями, которые дают концы интервала величин.

Выявляется итеративная форма алгоритма, в которой появляются так-называемые суммы со звездочкой.

Применение предлагаемой организации и алгоритма SA-V рекомендуется для баз данных с небольшой летучестью, на которых часты заказы нахождения по интервалам величин.

4.IV.1974