[D, 1999] Durr Ch., Chrobak M., Reconstructing hv-convex polyominoes from orthogonal orojections. Information Processing Letters 69 (1999) pp. 283-291.

[R, 1966] H. J. Ryser. Combinatorial Mathematics, 1966.

[H, 1997] G.T. Herman and A. Kuba, editors. Discrete Tomography: Foundations, Algorithms and Applications. Birkhauser, 1999.

[B, 1996] E. Barcucci, A. Del Lungo, M. Nivat, and R. Pinzani. Reconstructing convex polyominoes from horizontal and vertical projections. Theoret. Comput. Sci., 155:321{347, 1996.

[W, 2001] G.J. Woeginger. The reconstruction of polyominoes from their orthogonal projections. Inform. Process. Lett., 77:225{229, 2001.

Author's Information

Hasmik Sahakyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am

ALGORITHMIC MINIMIZATION OF NON-ZERO ENTRIES IN 0,1-MATRICES

Adriana Toni, Juan Castellanos, Jose Erviti

Abstract: *In this paper we present algorithms which work on pairs of 0,1- matrices which multiply again a matrix of zero and one entries. When applied over a pair, the algorithms change the number of non-zero entries present in the matrices, meanwhile their product remains unchanged. We establish the conditions under which the number of 1s decreases. We recursively define as well pairs of matrices which product is a specific matrix and such that by applying on them these algorithms, we minimize the total number of non-zero entries present in both matrices. These matrices may be interpreted as solutions for a well known information retrieval problem, and in this case the number of 1 entries represent the complexity of the retrieve and information update operations.*

Keywords*: zero-one matrices, analysis of algorithms and problem complexity, data structures, models of computation*

Introduction

We introduce some notation and concepts that will be useful from now on.

Let $I_{i,j}^m$ denote the matrix resulting from permuting the i^{th} and j^{th} rows in the identity matrix of dimensions m × m, denoted I^m. For any matrix *M* of dimensions $m \times n$, $I_{i,j}^m \times M$ returns the matrix *M* in which rows *i, j* have switched position.

Generally, if $I_\sigma^m = I_{i_1,j_1}^m \times I_{i_2 j_2}^m \times \ldots \times I_{i_k,j_k}^m$, the effect of the multiplication $I_\sigma^m \times M$ is to switch the position of rows i_k and j_k of *M*, *then* do the same thing with rows i_{k-1} and j_{k-1}, *then* with rows i_{k-2} and j_{k-2}… until finally rows i_1, j_1 have been switched.

Let H be the matrix of dimensions $\frac{n(n+1)}{2} \times n$ *defined by:*

$$H_{ij} = \begin{cases} 1 & l \le j \le l + (i - w_{l-1} - 1) \\ 0 & otherwise \end{cases}$$

where

$w_k = \sum_{l=0}^{k-1}(n-l)$, $k=0\ldots n$

$i \in \{(w_{l-1}+1)\ldots w_l\}$, $l=1\ldots n$

Remark 1

Note that if T^n is the triangular matrix of dimensions $n \times n$ consisting of 0s above the main diagonal and 1s along and below the main diagonal,

$$T=(t_{ij})_{i,j=1..n} \quad \text{with} \quad t_{ij}=\begin{cases}1 & i \geq j\\ 0 & i \prec j\end{cases}$$

then the following statement holds for all k = 0…(n-1),

$$H_{[w_k+1..w_{k+1}],[k+1..n]} = T^{n-k}$$
$$H_{[w_k+1..w_{k+1}],[1..k]} = 0$$

where $w_i = \sum_{s=0}^{i-1}(n-s)$ *for i = 0…n, and* $H_{[i\ldots j],[r\ldots s]}$ *is the box of matrix H composed of the rows in the interval [i…j] and the columns in the interval [r…s].*

Hence, for any given dimension n, the corresponding matrix H^n can be expressed as a function of the matrix T of different dimensions as:

$$H = \begin{pmatrix} T^n \\ \overrightarrow{T^{n-1}} \\ \vdots \\ \overrightarrow{T^{n-(n-1)}} \end{pmatrix} \quad \text{where} \quad \overrightarrow{T^{n-i}} = \begin{pmatrix} 0 & \ldots & \ldots & 0 & \\ \vdots & & & \vdots & \\ \vdots & & & \vdots & \bigg| T^{n-i} \\ 0 & \ldots & \ldots & 0 & \end{pmatrix} \quad i=1..(n-1)$$

Example 2

The matrix H of sizes n=2 and n=4, denoted H^2 and H^4, respectively, is shown below. Lines have been added to highlight the logical division into boxes

$$H^2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ \hline 0 & 1 \end{pmatrix}, \qquad H^4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{pmatrix}$$

For any $n = 2^k$, any row rearrangement of matrix H^n can be achieved by multiplying H^n by a certain identity transform I_σ, where $\sigma \in Permutations\left(\left(\frac{n+1}{2}\right)\right)$ and $Permutations(k) = \{\ f : \{1...k\} \rightarrow \{1...k\}$/ f one to one-onto, $k \in N^+\}$. This is due to a known algebraic result, stating that any permutation that is a member of *Permutations(k)* can be expressed as a composition of a certain number of permutations of that set whereby all the elements of {1…*k*} save two are held fixed.

The matrix H^n should ultimately be rearranged as the matrix S^n, which is defined below.

Definition 3 *For all n of the form 2^l with $l \in N^+$, let*
$$S_n = \begin{pmatrix} H^{\frac{n}{2}} & 0 \\ M_1^{\frac{n}{2}} & T^{\frac{n}{2}} \\ M_2^{\frac{n}{2}} & T^{\frac{n}{2}} \\ \vdots & \vdots \\ M_{\frac{n}{2}}^{\frac{n}{2}} & T^{\frac{n}{2}} \\ 0 & H^{\frac{n}{2}} \end{pmatrix}$$

where for any $m \in N^+$, k=1…m, the matrices M_k^m are square matrices of dimensions m defined as
$$(M_k^m)_{i,j} = \begin{cases} 1 & k \le j \\ 0 & otherwise \end{cases}$$

Hereinafter let $I_{H^n \to S^n}$ denote the matrix I_σ that leads to the transformation of H^n into S^n.

Hence, $I_{H^n \to S^n}$ x $H^n = S^n$.

Let us look at a couple of simple examples from which the specific expression of the matrix $I_{H^n \to S^n}$ can be easily deduced.

Example 4 *If* n=2, *then* $H^2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ \hline 0 & 1 \end{pmatrix} = S^2$ *and hence* $I_{H^2 \to S^2} = I^2$

Let us also look at the example of the transformation of H^4 into S^4, illustrating the operations that need to be performed and the expression corresponding to the matrix $I_{H^4 \to S^4}$.

Here

$$H^4 = \begin{pmatrix} 1&0&0&0 \\ 1&1&0&0 \\ 1&1&1&0 \\ 1&1&1&1 \\ \hline 0&1&0&0 \\ 0&1&1&0 \\ 0&1&1&1 \\ \hline 0&0&1&0 \\ 0&0&1&1 \\ \hline 0&0&0&1 \end{pmatrix} \xrightarrow{I_{3,5}} \begin{pmatrix} 1&0&0&0 \\ 1&1&0&0 \\ 0&1&0&0 \\ 1&1&1&1 \\ \hline 1&1&1&0 \\ 0&1&1&0 \\ 0&1&1&1 \\ \hline 0&0&1&0 \\ 0&0&1&1 \\ \hline 0&0&0&1 \end{pmatrix} \xrightarrow{I_{4,5}} \begin{pmatrix} 1&0&0&0 \\ 1&1&0&0 \\ 0&1&0&0 \\ 1&1&1&0 \\ \hline 1&1&1&1 \\ 0&1&1&0 \\ 0&1&1&1 \\ \hline 0&0&1&0 \\ 0&0&1&1 \\ \hline 0&0&0&1 \end{pmatrix} = S^4$$

Hence, $I_{H^4 \to S^4} = I_\sigma = I_{4,5} \times I_{3,5}$

Where $\sigma : \{1..10\} \to \{1..10\},\quad \sigma(3) = 4,\ \sigma(4) = 5,\ \sigma(5) = 3$

Next we introduce a well known information retrieval problem and define a matricial model for the design of solutions and the study of operations complexity. In this model, the information retrieve and update operations are represented by 0,1-matrices. We will define pairs of matrices and the corresponding algorithms on them which implement these operations, and will apply on them algorithms which reduce the number of non-zero entries, as in this model, the operations complexity is defined as the number of 1s present in both matrices.

Let $v_1..v_n$ be variables storing values from an arbitrary commutative semigroup S. We desire to execute the following operations on these variables:

a) retrieve(i,j) returns $v_i + \ldots + v_j$ $\quad \forall\, 1 \le i \le j \le n$,

b) update(i,x): $v_i := v_i + x$ $\quad \forall\, 1 \le i \le n, \quad x \in S$

This problem is known as the range query problem of size n.

We can organize the variables as an array V of length n, and implement the operations as above. In this case, the complexity of executing an update operation is constant meanwhile the worst case complexity of a retrieve is linear on n.

Our interest centers upon improving the average complexity of the operations assuming that each one of them is selected with the same probability.

We can use different data structures involving a different number of variables storing values in the semigroup, and provide the corresponding algorithms to implement the update and retrieve operations, and still be solving the same computational problem.

A matricial model for the study of the range query problem has been defined, relative to which computational complexity is assesed (see [6])

The model comprises all programs verifying:

a) A set of variables $Z=\{z_1, z_2..z_m\}$is maintained.

b) Retrieve(i,j) is performed by adding up a subset of these variables.

c)Update(j,x) is performed by incrementing a subset of these variables by amounts which depend linearly on x .

The model defined in [6] consists of triples <R,U,Z> where

$Z = \{z_1 \ldots z_m\}$ is a set of variables storing values on an arbitrary semigroup S, $R=(r_{i,j})$ is a zero-one matrix of dimension $\frac{n(n+1)}{2} \times m$ and $U=(u_{i,j})$ is a zero-one matrix of dimension $m \times n$. Each row of R describes the subset of variables of Z which have to be added to execute one of the retrieve operations, and the *i-th* column of U describes the subset of such variables which have to incremented to execute an update(i,x). So, a pair of R and U matrices describes a solution for the range query problem of size n (m, the number of required program variables, may change although if has to be greater or equal n). Associated with a triple <R,U,Z>, the programs implementing the operations are defined as follows.

Definition 5

Given a triplet <Z,R,U> within the matrix model for the range query problem of size n, with $Z = \{z_1 \ldots z_m\}$, *then the update and retrieve operations must be implemented through the following programs:*

- *update(j,x): for l:=1 to m do* $[z_1 \leftarrow z_l + u_{l,j}\, x]$
- *Retrieve(i,j): output* $\sum_{l=1}^{m} r_{k,l} z_l$, *where* $k = \sum_{s=0}^{i-2} (n-s) + (j-i+1)$

The following proposition establishes a condition on *R, U* that entails reworking the programs defined above.

Proposition 6

Let H be the matrix of dimensions $\frac{n(n+1)}{2} \times n$ *defined as at the beginning of the introduction section.*

Then the programs given in Definition 5 represent a solution for the range query problem of size n if and only if $R \times U = H$.

In the following we define the complexity associated with the operations within the matrix model.

Definition 7

Given a triplet <Z,R,U> that solves the range query problem of size n within the matrix model, with Z = ($z_1 \ldots z_m$), $R = (r_{i,j})_{i=1\ldots\frac{n(n+1)}{2}, j=1\ldots m}, U = (u_{i,j})_{i=1\ldots m, j=1\ldots n}$, *let the complexity associated with the Retrieve (i, j) operation be defined as:*

$$|\{r_{kl} / r_{kl} \neq 0 \wedge (1 \leq l \leq m)\}| \quad \text{where} \quad k = (j - i + 1) + \sum_{s=0}^{i-2}(n - s)$$

and the complexity associated with the Update (j, x) operation be defined as:

$$|\{u_{lj} / u_{lj} \neq 0 \wedge (1 \leq l \leq m)\}|$$

Let *m* be the number of columns of *R* and of rows of *U.* The average complexity of *Update* operations is given by

$$p = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}}{n}$$

and the average complexity of the *Retrieve* operations by

$$t = \frac{\sum_{i=1}^{n(n+1)/2}\sum_{j=1}^{m} r_{ij}}{n(n+1)/2}$$

It is known for any data structure solving the *range query problem of* size *n* that $\rho + t = \Omega(\log n)$

Range query Problem-Solving Matrices

In the following we will define pairs of matrices of 0s and 1s whose product is the matrix *H* —that is, matrices that represent solutions to the *range query problem* – in an attempt to minimize the total number of 1s present in both matrices and, therefore, the average complexity of the operations. A recursive definition will be given later (see Definition 13). In particular, let the matrices $R = (r_{i,j})$, $U = (u_{i,j})$ be defined, whose product for any dimension *n* of the form 2^k with $k \in N^+$ is H^n.

Remember that the average complexity is calculated by dividing the total number of 1s by the number of different operations. Hence, if we are dealing with the problem of size *n* and let $\psi(n) = \sum_{i=1}^{n(n+1)/2}\sum_{j=1}^{m} r_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}$

where *m* is the number of variables $z_1 \ldots z_m$ used to implement the solution (*m* may vary, although it necessarily has to be greater than or equal to *n*), then the average complexity is given by $\frac{\psi(n)}{n + \frac{n(n+1)}{2}}$ (*n* is the number of different *Update(j, x)* operations as a function of the first argument and $\frac{n(n+1)}{2}$ is the number of different possible arguments for a *Retrieve(i, j)* operation).

We will prove that our matrices hold for

$$\psi(n) = \frac{3}{2}n^2 - \frac{3}{2}n\log_2 n + \frac{9}{2}n - 2\log_2 n - 4$$

and this implies an average complexity that has a constant order of complexity.

Remember that a superindex is used to specify the size of the problem corresponding to the matrix *H*. Hence H^n denotes the matrix for the *range query problem of siz*e n, although the size of H^n is $\frac{n(n+1)}{2} \times n$.

Remark 8

How does this type of transformations *affect the matrix approach to the range query problem? We know that it involves studying pairs of integer matrices R^n, U^n such that $R^n \times U^n = H^n$. But if $R^n \times U^n = H^n$, then $I_{H^n \to S^n}$ x $R^n \times U^n = S^n$. Hence the problem can be reformulated equivalently as entailing the study of matrix pairs whose product is the matrix S^n. In this case the algorithm that implements the Retrieve operations given in Definition 1 has to be modified, and the definition of the programs associated with a triplet <Z,R,U> is now as follows:*

Definition 9

Given a triplet <Z,R,U>, with $Z = (z_1 \ldots z_m)$, R, U matrices of dimensions $n \times m$ and $m \times n$, respectively, let us define the following algorithms to implement the Update and Retrieve operations:

1. *Update(j,x): for l: 1 to m do* $z_l \leftarrow z_l + u_{l,j} x$
2. *Retrieve(I,j): output* $\sum_{l=1}^{m} r_{k,l} z_l$ *where k is given by:*

   a) $k = \sum_{s=0}^{i-2} \left(\frac{n}{2} - s \right) + (j - i + 1), \qquad 1 \le i \le j \le \frac{n}{2}$

   b) $k = \sum_{s=0}^{i-2} (n - s) + (j - i + 1), \qquad \frac{n}{2} < i \le j \le n$

   c) $k = \frac{n}{2}(i-1) + \left(j - \frac{n}{2} \right) + \frac{\frac{n}{2}\left(\frac{n}{2} + 1 \right)}{2}, \qquad i \le \frac{n}{2}, \qquad j \ge \frac{n}{2}$

The intuitive idea is that the row number (k) of the matrix R^n associated with a Retrieve(i,j) operation is different now, as some rows have switched position.

The change of approach has no bearing on the complexity study of the operations, as, remember, the effect of multiplying any matrix by a certain I_σ does not alter the number of non-null matrix elements, but only switches the position of certain rows.

Recursive Definition of Our Problem-Solving Matrices

In this section we will give a recursive definition of our matrix pairs R^n, U^n as a function of the problem size *n*. The matrices hold for $R^n \times U^n = H^n$.

As mentioned already, the definition is valid for values of the form $n = 2^k$.

Let us refer to *blocks* of consecutive rows of the matrix R^n, which we consider to be divided into *n* horizontal blocks, the first formed by the first *n* rows, the second by the next (*n-1*) rows, the third by the (*n-2*) rows… up to the *(n-1)*th block, which is composed of two rows and the n^{th} block which consists of just the last row.

Let R_i^n denote the i^{th} block of R^n and $R_i^n(j)$, the j^{th} row of this block such that the matrix R^n is given by

$$R^n = \begin{pmatrix} R_1^n \\ R_2^n \\ \vdots \\ R_n^n \end{pmatrix}$$

(note that if the dimensions of R^n are $\frac{n(n+1)}{2} \times m$, then the dimensions of each block R_i^n are $(n-i+1) \times m$).

R^n, U^n pairs are constructed by applying a function called *Refinement*. This function can be viewed as a two-stage process: the first stage involves executing a sequence of extension steps, and the second rearranging the rows of R^n by multiplying by a given identity transform matrix I_σ.

The following Definition 11 and Lemma 11 are needed to define the extension step concept.

Definition 10

Given a matrix M of 0s and 1s, two columns i, j are said to be disjoint if the set of rows $\{k/m_{k,i}=1=m_{k,j}\}$ *is empty. Similarly, two rows i, j of M are said to be disjoint if the set of columns* $\{k/m_{i,k}=1=m_{j,k}\}$ *is empty.*

Lemma 11

Let A, B be two matrices of 0s and 1s such that $A\times B = S^n$, *of dimensions* $\frac{n(n+1)}{2}\times m$ *and* $m\times n$, *respectively. Assume that there are two columns i, j of A that are not disjoint and let* $\{l_1\ldots l_q\}$ *be the set of rows of A for which* $a_{l_k,i}=1=a_{l_k,j}$, *k=1…q holds. Then the rows i, j of B are disjoint.*

Definition 12

Let A, B be two matrices of 0s and 1s such that $A\times B = S^n$, *of dimensions* $\frac{n(n+1)}{2}\times m$ *and* $m\times n$, *respectively.*

Assume that there is a set of columns $C=\{c_1\ldots c_l\}$, $l\geq 2$, *for which there is a non-empty* maximal *set – including all the rows that meet the following condition – of rows* $F=\{f_1,\ldots,f_q\}$ *such that* $a_{f_i,c_j}=1\quad \forall c_j\in C,\ \forall f_i\in F$

We define the extension step associated with the sets C, F as the execution of the following actions on A and B:

1. *Insert a new column* z_0 *in A such that* $a_{i,z_0}=1 \Leftrightarrow i\in F \qquad \forall i=1..\frac{n(n+1)}{2}$
2. *Add a new row* z_0 *in B such that* $b_{z_0,j}=\sum\{b_{k,j}/k\in C\} \qquad \forall j=1..m$
3. *Columns* $c_1\ldots c_l$ *of A are modified as follows* $a_{f_i,c_k}:=0 \qquad \forall c_k\in C,\quad \forall f_i\in F$

It can be easily demonstrated that if an extension step is applied to a matrix pair whose product is the matrix S^n, *the product of the resulting matrices is the very same matrix* S^n.

We are now able to give a recursive definition of our matrix pairs.

Definition 13

Let us recursively define matrix pairs R^n, U^n *of dimensions* $\frac{n(n+1)}{2}\times m$ *and* $m\times n$, *respectively, with n* of the form 2^k with $k\in N^+$, *as follows*

1. *n=2:* $\langle R^2,U^2\rangle=\langle H^2,I^2\rangle$
2. *n=*2^k*:* $\langle R^2,U^2\rangle$*=Refinement(*$\hat{R}^{2\hat{n}}$, $\hat{U}^{2\hat{n}}$*) where*

   a) $$\hat{R}^{2n}=\begin{pmatrix} R^n & 0 \\ f_n^m(R_1^n(n)) & R_1^n \\ f_n^m(R_2^n(n-1)) & R_1^n \\ \vdots & \vdots \\ f_n^m(R_n^n(1)) & R_1^n \\ 0 & R^n \end{pmatrix},\quad \hat{U}^{2n}=\begin{pmatrix} U^n & 0 \\ 0 & U^n \end{pmatrix}$$ *(m is the number of columns of* R^n*)*

b) $f_k^m : \{0,1\}^m \to [R^m \to R^k]$, *that is to say* $f_k^m(v_1..v_m)$, *returns a matrix —a linear mapping— of dimensions k × m.* f_k^m *is defined such that the k rows are precisely the argument vector (v1…vm):* $f_k^m(v_1..v_m) = \begin{pmatrix} v_1 & \dots & \dots & v_m \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ v_1 & \dots & \dots & v_m \end{pmatrix}$

c) *Refinement(* $\hat{R}^{2n}, \hat{U}^{2n}$ *) is a two-phase process.*

i. Extension steps: as many consecutive extension steps are executed on the matrix $\hat{R}^{\hat{2}\hat{n}}$ *as necessary to assure that each row of the blocks* $f_n^m(R_i^n(n-i+1)), i = 1...n$, *and the blocks* R_1^n *have just one 1. The extension steps should be bound to sets of columns C that include either columns of the left-hand blocks* only*—blocks of the form* $f_n^m(R_i^n(n-i+1))$*— or columns of the right-hand blocks —of the form* R_1^n.

Let $\hat{R}'^{2n}$, $\hat{U}'^{2n}$ *denote the matrices resulting from executing these extension steps.*

The matrix $\hat{U}'^{2n}$ *is* ***actually the final matrix*** U^{2n} *that we aim to define, as it is unaffected by the second phase of the Refinement process.*

Note: We have proved that the number of extension steps needed to construct R^{2n}, U^{2n} *is exactly* $2\left(\frac{n}{2} + \left(\frac{n}{2^2} - 1\right) + \left(\frac{n}{2^3} - 1\right) + ... + 1\right)$.

ii. Rearrangement: the product of the matrices $\hat{R}^{2n}, \hat{U}^{2n}$ *is a matrix* S^{2n} *from which it likewise follows that the product of the matrices* $\hat{R}'^{2n}$, $\hat{U}'^{2n}$ *is* S^{2n}*, as it has already been demonstrated that the extension steps do not affect the product of the matrices. Assuming this result, this phase involves rearranging the matrix* $\hat{R}'^{2n}$ *by means of the multiplication* $I_{S^{2n} \to H^{2n}} \times \hat{R}'^{2n}$ *, where* $I_{S^{2n} \to H^{2n}}$ *is the matrix that holds for* $I_{S^{2n} \to H^{2n}} \times S^{2n} = H^{2n}$ *. Finally, the matrix* R_1^n *that we are trying to define is precisely* $R^{2n} = I_{S^{2n} \to H^{2n}} \times \hat{R}'^{2n}$.

Note: the existence of the matrix $I_{S^{2n} \to H^{2n}}$ *is straightforwardly deduced from the existence of the matrix* $I_{H^{2n} \to S^{2n}}$ *, since if the expression corresponding to* $I_{H^{2n} \to S^{2n}}$ *is* $I_{i1,j1}^c \times I_{i2,j2}^c \times ... \times I_{ik,jk}^c$ *, where c is the number of rows of* H^{2n}*, then* $I_{S^{2n} \to H^{2n}}$ = $I_{ik,jk}^c \times ... \times I_{i2,j2}^c \times I_{i1,j1}^c$.

Remarks 14

a) From the definitions of the matrices T^n, S^{2n}, H^{2n} $\hat{R}^{2n}, \hat{U}^{2n}$ *it follows that* $\hat{R}^{2n}$ *x* $\hat{U}^{2n}$ *=*S^{2n}

b) As a consequence, and by definition of R^n, U^n*, it holds that* $R^n \times U^n = H^n$.

c)The maximum number of 1s present in each row of R^n *is 2, whatever the value of n.*

d)Let $n = 2^{k+1}$ *for a certain natural number k. The number of extension steps that are executed in the Refinement phase of the matrices construction process is* $2\left(\frac{n}{2} + \left(\frac{n}{2^2} - 1\right) + \left(\frac{n}{2^3} - 1\right) + ... + 1\right)$.

e)Let A, B be two matrices of 0s and 1s, such that A × B = S^n*, of dimensions* $\frac{n(n+1)}{2} \times m$ *and m × n, respectively.*

The execution of an extension step associated with the column and row sets C= $\{c_1 \ldots c_l\}$, F=$\{f_1 \ldots f_q\}$ of A, respectively, leads to a change in the total number of 1s present in the two matrices according to the following expression:

$\sum_{i\in C, j\in\{1..n\}} b_{i,j} + q - (l \times q)$. *If the value of* $\sum_{i\in C, j\in\{1...n\}} b_{i,j} + q - (l \times q)$ *is greater than 0 then the total number of 1s in the matrices increases; if the value is equal to 0 then the number of 1s is unchanged, and if the value is less than 0 the total number of 1s decreases.*

As a consequence, each extension step executed in the Refinement phase of the process of constructing our matrices given in Definition 13 decreases the total sum of the number of 1s present in the two matrices.

Theorem 15

Let R^n, U^n be matrices of dimensions $\frac{n(n+1)}{2} \times m$ *and $m \times n$, respectively, with n* of the form 2^k with $k \in N^+$, *as defined in Definition 13.*

Let $\#R^n$, $\#U^n$ denote the number of 1s in the matrices R^n and U^n respectively.

It holds that

$$\#R^n + \#U^n = \frac{3}{2}n^2 - \frac{3}{2}n\log_2 n + \frac{9}{2}n - 2\log_2 n - 4$$

This represents a constant average complexity for the set of $\frac{n(n+1)}{2} + n$ *Retrieve and Update operations.*

As regards the number of variables $z_1 \ldots z_m$ required by the solution defined by our matrices as a function of the problem size *n*. Let *Var(n)* denote this number of variables, which, as we know, is the same as the number of columns and rows of R^n *and* U^n respectively. *It holds that*

$$m = n\log_2 n - 2n + 2\log_2 n + 2$$

Proof

For the proof of these results, see [5].

Bibliography

[1] D.J. Volper, M.L. Fredman, *Query Time Versus Redundancy Trade-offs for Range Queries*, Journal of Computer and System Sciences 23, (1981) pp.355--365.

[2] W.A. Burkhard, M.L. Fredman, D.J.Kleitman, *Inherent complexity trade-offs for range query problems*, Theoretical Computer science, North Holland Publishing Company 16, (1981) pp.279--290.

[3] M.L. Fredman, *The Complexity of Maintaining an Array and Computing its Partial Sums*, J.ACM, Vol.29, No.1 (1982) pp.250--260.

[4] A. Toni, *Lower Bounds on Zero-one Matrices*, Linear Algebra and its Applications, 376 (2004) 275--282.

[5] A. Toni, Complejidad y Estructuras de Datos para e*l problema de los rangos variables*, Doctoral Thesis, Facultad de Informática, Universidad Politécnica de Madrid, 2003.

[6] A. Toni, *Matricial Model for the Range Query Problem and Lower Bounds on Complexity*, submitted.

Authors' Information

Adriana Toni – atoni@fi.upm.es

Juan B. Castellanos – jcastellanos@fi.upm.es

Jose Joaquin Erviti – jerviti@fi.upm.es

Facultad de Informática, Universidad Politécnica de Madrid, Spain